

Preface

The Workshop “From Objects to Agents” (WOA) is the reference event for Italian researchers active in the Agents and Multi-Agent Systems research domain. Since its very first edition in 2000, located in Parma (Italy), WOA was conceived as a meeting occasion for researchers and practitioners from the working group on MAS of AI*IA and from the TABOO association (Advanced Technologies Based on Concepts from Object-Orientation). After that, WOA was held on a yearly basis in many different Italian locations, from north to south, gaining a conspicuous success and succeeding in gathering researchers and practitioners from various research fields, thanks to its format.

Despite stemming from an Italian initiative, WOA is an international workshop where presenters and participants exchange opinions and discuss on-going works in a friendly yet rigorous setting. Furthermore, since 2004, WOA includes a one-day mini-school, where experienced scientists and professionals can introduce younger researchers as well as Ph.D. and undergraduate students to hot topics in the fields of AI, MAS, and Programming Languages.

The 22nd edition of the workshop has been held on September 1–3, 2021 in Bologna. During these three days, more than 18 speakers joined the workshop, as well as many more listeners. In particular, this edition was structured in two mini-school sessions, one keynote speech, and in five technical sessions. The six technical sessions hosted the presentation of 17 papers collected in this virtual volume published by CEUR.

The topics discussed in the papers covered some of the hottest topics laying under the umbrella of “Multi-agent systems in the machine learning era”, as requested by the call for papers. The choice of this theme was deliberate. In fact, it has been widely recognised that artificial intelligence is here to stay, along with both its symbolic and sub-symbolic branches. While symbolic techniques played a central role in the early days of AI, the last decade has been characterised by the explosion of data-driven, machine-learning-based intelligent systems. Nowadays, the exploitation of machine learning (ML) and big data processing is so pervasive that novel results and applications are being proposed at an unprecedented pace.

While multi-agent systems (MAS) have been entangled with symbolic AI since the very beginning, it is undoubted that they have been extensively exploited in several data-intensive domains including – but not limited to – robotics, telecommunications, simulation, decision support systems, and economics. MAS and ML have the potential to benefit from each other. In fact, on the one side, it is becoming increasingly complex for agent designers to forecast all possible operating situations, so learning from data is fundamental for next-generation MAS—in order to adapt to the intricacies of an increasingly complex world. On the other side, modern ML-based solutions may easily struggle when different data-driven solution – possibly attained from disparate data sources – must be integrated, combined, reconciled or explained. Under that perspective, MAS have certainly a role to play to advance the state of the art of data-driven AI towards integration and explainability.

The intertwining between MAS and ML, therefore, aims at bringing about huge benefits to the field of data-driven AI, both at the theoretical and the practical level; however, a well-grounded integration in terms of methodologies and systems properties is far from being reached and acknowledged.

As far as the mini-school is concerned, two sessions were organised, hosting talks from experts in the fields of Logic Programming and MAS. In particular, in the first session, Matteo Baldoni discussed the history of MAS, from objects to agents. The talk introduced autonomous agents and multi-agent systems, defining the meaning of intelligence in the context of intelligent agent systems, and discussing the difference between objects and agents, focusing on the relationship between the object-oriented and the agent-oriented paradigm. The lecture ended with some remarks about JaCaMo.

In the second session, Marco Gori provided a perspective on the current status of ML and MAS, mostly based on his recent book “Machine Learning: A Constraint-Based Approach”. The talk provided participants with a refreshing look at the basic models and algorithms of machine learning – with a glance to MAS –, emphasising current topics of interest including neural networks and kernel machines.

The “Fabio Bellifemine” keynote speech was given by Antonio Lieto, who discussed the topic of “Bounded and Resource-Rational Agents for Integrated Intelligence”. There, the speaker presented the different notions of rationality as they are developed in the field of cognitive science – i.e., classical rationality (CR), bounded rationality (BR), resource-rationality (RR) – and discussed their impact on the design and implementation of intelligent systems. In particular, Lieto argued that, in order to build integrated AI systems able to exhibit a wide range of intelligent behaviours, it is crucial to take into account bounded-rational and resource-rational cognitive constraints. In doing so, two cognitively inspired AI applications – Dual PECCS and the TCL reasoning framework – were presented, showing how the outlined design perspective allowed such systems to (i) address some crucial aspects of commonsense reasoning in AI research (namely, dealing with typicality effects and with the problem of commonsense compositionality), (ii) integrate those systems with more general cognitive architectures, (iii) use their simulations as “computational explanations” to better understand the heuristics used by the human mind to face complex problems.

The 17 papers collected in this issue were presented and discussed into five thematic sessions. The final versions here included also include the outcomes of some of the discussions that followed the presentations at the workshop. The authors’ contributions cover extremely relevant research areas that include (i) trustworthy & explainable MAS, (ii) MAS & subsymbolic AI, (iii) logic in MAS, (iv) agent-based modelling & simulation, (v) language, tools & application.

In the end, the Organising Scientific Committee gratefully thanks all those who, with their work and their enthusiasm, have contributed to the success of this edition of WOA: the members of the Program Committee, the Department of Informatics Engineering and Information Sciences (DISI) of the University of Bologna, the Alma Mater Research Institute for Human-Centered Artificial Intelligence of the University of Bologna, the local organisers, the speakers of the workshop sessions, the mini-school lecturers, the sponsors, and all collaborators who participated in the organisation. Overall, they would like to thank the lively, creative and sometimes volcanic community that has been regularly meeting for 22 years at the workshop.

Roberta Calegari, Giovanni Ciatto, Enrico Denti, Andrea Omicini, and Giovanni Sartor

Architectural Technical Debt of Multiagent Systems Development Platforms

Ilaria Pigazzini¹, Daniela Briola¹ and Francesca Arcelli Fontana¹

¹Department of Informatics, Systems and Communication (DISCO), University of Milano - Bicocca, 20125 Milan, Italy

Abstract

Technical debt is candidate to be the next *buzzword* in software engineering, and the number of studies evaluating the technical debt of software projects is increasing. A particular and dangerous type of debt is the architectural debt, i.e., the consequences of sub-optimal design decisions. Currently, there are no studies about the evaluation of architectural debt in MultiAgent Systems (MAS) and platforms. Hence, in this paper we propose the analysis of four well-known MAS development platforms, with the aim of evaluating their architectural debt and open the discussion in this field. We exploit a tool, named Arcan, developed for architectural smell detection and for the computation of an architectural debt index. The results show that MAS development platforms are subjected to architectural debt, and in particular to the presence of Cyclic Dependency smells. However, there is evidence that the minimum amount of debt is reached when developers report “bug fixes” and “Improvements”.

Keywords

architectural debt, architectural smells, multiagent system platforms, trend analysis

1. Introduction

Technical Debt (TD) is “a metaphor reflecting technical compromises that can yield short-term benefit, but may hurt the long-term health of a software system” [1]. Architectural Technical Debt (ATD) is a specific type of TD limited to the architecture (design decisions) of a software system [1] and is considered as the most dangerous and critical one [2]. Systems affected by ATD are hard to maintain and evolve.

The concept of TD is not recent [3], however the research has been active especially in the past few years. Works on TD and ATD have been done on monolithic systems [4][5], distributed systems such as microservices [6], machine learning systems [7] and also on IoT systems [8]. However, to the best of our knowledge, there are no study about ATD in MultiAgent Systems (MAS) and MAS development platforms. Anyway, the MAS community is deserving to reliability, scalability and in general Software Engineering (SE) aspects more and more attention in the last years, as confirmed for example by the creation of the dedicated SE area of interest at AAMAS (International Conference on Autonomous Agents and Multiagent Systems International Conference on Autonomous Agents and Multiagent Systems), workshops focusing on SE topics (for example EMAS (Engineering Multi-Agent Systems) and AREA (Agents and Robots for reliable Engineered Autonomy) [9]), and works on Engineering MultiAgent

WOA 2021: Workshop “From Objects to Agents”, September 1–3, 2021, Bologna, Italy

✉ i.pigazzini@campus.unimib.it (I. Pigazzini); daniela.briola@unimib.it (D. Briola); i.tiddi@vu.nl (F. A. Fontana)

🆔 0000-0003-2629-6762 (I. Pigazzini); 0000-0003-1994-8929 (D. Briola); 0000-0002-1195-530X (F. A. Fontana)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Systems or surveying in a systematic way the available technologies (for example [10, 11, 12]): so, it is only a matter of time before other SE topics will be faced by the MultiAgent Systems community too.

In this paper, we aim to analyse four well-known and largely adopted MAS development platforms (Jade, Jason, Jadex and Netlogo) in order to evaluate their architectural debt: since these platforms are used by many developers and have been released in many versions in a quite long lifespan, we are interested in evaluating if they suffer of ATD, so that in case to provide to their developers useful hints to improve their quality.

We exploit Arcan [13], our tool for Architectural Smells (AS) detection and ATD estimation. In particular, we compute the Architectural Debt Index (ADI) [4], which is a value indicating the amount of architectural technical debt present in a project, based on the AS that affect it [13]. AS (which by some authors are referred to as anti-patterns) are design decisions which impact negatively on the internal quality of software systems and in this study we consider three different types of AS based on dependency issues.

Our results show that the considered systems suffer from ATD, thus their developers should be aware of it so that to be able to manage these issues in future release.

The outline of our paper is the following: Section 2 reports some related works regarding the quality of MAS platforms; Section 3 describes the study design, with the study research questions and the analysis we conducted to answer them; Section 4 reports the results of our analysis and the answers to the research questions; finally Section 5 contains the conclusions and future developments of our work.

2. Related Works

For the best of our knowledge, this is the first study aiming at evaluating the architectural debt of MAS platforms. In general, we found few references to the quality of MAS platforms. However, even if we did not find studies about their maintainability and evolvability (the two quality attributes most impacted by ATD) we found some works about the security, performance and scalability of MAS platforms. We observed that the quality of the design aspects of MAS platforms and MAS systems are not yet often taken in consideration and discussed [14], even if they have an impact on their final performance, probably because the community is still more focus on the previous mentioned aspects which prevent whatever platform to be concretely and largely adopted.

Concerning security, Endsuleit et al. [15] performed a security analysis on the multiagent platform Jade in its version 3.2 as well as on its security plugin Jade-S. They reported a classification of possible and well-known attacks on Jade and provided a discussion on what is still missing in Jade-S. They also present some Denial-of-Service attacks which they have implemented and successfully tested.

Concerning the performance, Mulet et al. [16] investigated the relationship between performance (in terms of agents' response time to messages) and internal design, that is, to identify the key design decisions that lead to better performance. They measured the performance of three Open-Source MAS platforms, namely Jade, MadKit and AgentScape. They found out that design decisions related to the modularity of the platform, such as offering a message service

by means of agents instead of implementing it in the kernel, degrades performance. Moreover, centralizing services in a single host in the platform also degrades performance because the host can become a bottleneck in the case of very popular services.

A similar study was conducted by Alberola et al. [17], who analysed the same set of three MAS platforms and reached similar conclusions, i.e., that the design impacts the performance. In particular, they evaluated the response time of the three platforms when changing parameters like message traffic and the amount of agents running. They found that all the three platforms perform poorly and demonstrate low scalability when the MAS being run on increases.

To conclude, the field of MAS software quality and technical debt is not popular and researched yet, and with our work we aim to open the discussion about architectural debt and architectural smells by analysing the most used platforms for MAS development.

3. Study Design

We introduce the design of this study and the following Research Questions we aim to answer:

- *RQ1: Which is the most present type of AS in MultiAgents Systems platforms?*
- *RQ2: What can we observe according to architectural debt of MultiAgents Systems platforms?*

To answer the two RQs we evaluate the AD in terms of the AS and the Architectural Debt Index (ADI) computed through Arcan. Since we have large experience [13][4][18] in analyzing the AD in open source projects, but not in MAS development platforms, through the answer to these RQs we aim to analyze the AD of MAS platforms, in order to provide some preliminary hints to their developers. In case AD is present or specific AS are identified in the systems, developers have to pay attention to these problems to prevent them or remove them as soon as possible.

3.1. Analyzed projects

We selected four well-known MAS development platforms, namely Jade [19], Jadex[20], Jason[21] and Netlogo[22], and analysed their development history. These projects are written in Java, the programming language supported by Arcan. All projects but Jade are hosted on Github, which, given the large amount of code commits (code snapshots at specific points in time), enables the easy analysis of their history. Table 1 shows the main project characteristics: names, the number of analysed commits, the considered time period (date of the first and last commit), size expressed in Number of Lines of Code (LOC) both for the first and last commit and finally the download url. Concerning the commit analysis, we considered only commits pushed or merged into the master branch, starting from the beginning of the commit history and by sampling one commit every 30. We do not analyse each commit since architectural changes tend to happen in larger time spans with respect to code changes. A threat to such approach could be that by managing sampling by taking in consideration only the time gap, we would miss out “relevant commits”, where the architectural change actually happens. However, it is not important if we miss relevant commits, because we take into consideration the whole evolution and an

Table 1
Projects characteristics

Project	#Commits	First commit	Last commit	LOC first commit	LOC last commit	Download url
Jade	6	23/12/2015	06/06/2017	-	-	https://mavenrepository.com/artifact/com.tilab.jade/jade
Jadex	111	05/11/2008	16/03/2018	130288	502220	https://github.com/actoron/jadex
Jason	38	23/03/2017	20/04/2021	37447	45825	https://github.com/jason-lang/jason
Netlogo	25	05/08/2011	09/05/2016	60260	56075	https://github.com/NetLogo/NetLogo

architectural change happens during a span (not in a single commit). Similar custom samplings were used in similar context by previous studies [23, 24]. We conducted a different analysis for Jade, which is the only project not hosted on Github. We collected six versions from the Maven Central Repository¹ and run Arcan on all of them. Since we found only the jar files, we could not report the number of Lines of Code in Table 1.

Notice that the selected projects have different history, community, development team and purpose: indeed, different amounts of commits are sampled in each case. This makes the individual results difficult to compare directly. However, we propose a preliminary analysis which shall be complemented with manual validation from developers and additional context information. Section 3.3 offers the details of our analysis and how we compare the results among the different projects.

3.2. Collected Data

An architectural smell (AS) is a software design decision which negatively impact on the system internal quality, e.g., the system maintainability and ability to evolve. We collect data about the presence of AS because they are symptoms of architectural debt. AS can be of different types and have different side-effects.

We describe below the AS detected by Arcan considered in this work:

- *Unstable Dependency (UD)*: describes a component (package) that depends on other sub-systems that are less stable than the component itself. The components with an high instability are more prone to change with respect to the more stable ones, this means that the component which depends on less stable components is forced to change along with them.

¹<https://mavenrepository.com/artifact/com.tilab.jade/jade>

- *Hub-Like Dependency (HL)*: this smell arises when a component (class or package) has (outgoing and ingoing) dependencies with a large number of other components. The component affected by the smell is a unique point of failure and a dependency bottleneck. Moreover the logic inside a Hub-Like Dependency is hard to understand, and the smell causes change ripple effect.
- *Cyclic Dependency (CD)*: refers to a component (class or package) that is involved in a chain of relations that breaks the desirable acyclic nature of a system's dependency structure. The components involved in a CD can be hardly released, maintained or reused in isolation. Moreover, a change on one affected component will propagate towards all the other ones involved in the cycle.

Moreover, through Arcan we are able to compute the Architectural Debt Index of each project, which takes into account: (i) the *Number* of AS detected in a project, (ii) the *Severity* of an AS, where for Severity we mean the criticality of each instance of AS (an instance of a type of smell, such as CD, can be more critical with respect to another instance of CD smell) and (iii) the *Dependency metrics* of Robert Martin [25] (Instability, Fan In, Fan Out, Efferent, and Afferent Coupling) used for the AS detection. The higher the ADI value, the higher the debt. All the details about the ADI computation can be found in our previous work on this index [4].

We ran Arcan on the commits of each considered project and organized the results in a dataset, where each observation corresponds to a single commit of a single project. The columns of the dataset store the data about 1) the project the commit belongs to 2) the number of AS detected in the commit (one column for each type) and 3) the value of the ADI of the commit. The dataset and the analysis script are available in the replication package².

3.3. Analysis

In order to answer our research questions, we conducted two kinds of analysis on the dataset (number of AS and ADI). First, we extracted a **set of statistical metrics** (mean, standard deviation, minimum value, maximum value) for each project, to ease the interpretation of the Arcan analysis results. All metrics are evaluated with respect to the analysed time period, i.e., the data extracted from the considered commits. In this way, we can compare the statistics of the different projects, even if their ATD was evaluated on time periods of different length.

We also conducted **trend analysis** to understand how ADI and AS evolve overtime. We exploited the *Mann-Kendall test*, which is a non-parametric test able to assess if there is a monotonic upward or downward trend of the variable of interest over time. In our case, given the number of AS and the ADI value for each commit, the test is able to compare the values across history (i.e., the commits ordered by time of creation) and determine whether, along time, the number of AS and ADI increases/decreases or does not show a trend. If a trend is present, it can be the first clue that the presence of AS and ADI has a relationship with other kinds of variable, i.e., the maturity of the project, the seniority of the developers, the development practices adopted by the developers and so on.

Notice that this test can be used to find trends for as few as four samples. In our case, one sample corresponds to one commit. However, with only a few analysed samples, as in the

²<https://gitlab.com/essere.lab/public/mas-atd-evaluation>

case of Jade (only 6 versions), the test has a high probability of not finding a trend when one would be present if more commits were provided. Hence, we report also the results of Jade trend analysis, but knowing that they could be less relevant with respect to the other analysed projects.

Finally, we conducted a manual validation of the results of the tests. In particular, we collected the commit comments attached to Github and the available release notes. This was useful to offer an interpretation of the results of the single projects and to acquire information useful to compare the different projects among them.

4. Results

In this section, we report the results of our analysis and also the answers to our research questions. Table 2 reports results of the distribution analysis conducted on the four projects. The statistics are evaluated on the number of AS, also divided by AS type (CD, HL, UD), and on ADI, measured for each commit during the considered time period. The project with the highest mean number of AS is Jade (≈ 879), and it has also the highest mean value of ADI (≈ 38).

On the other hand, Netlogo has the lowest AS and ADI mean values. Notice that it is reasonable to have a non-zero number of AS in large projects as the considered ones. We analysed many Open Source Java projects in past works, indeed the ADI value is tuned with a reference dataset of past analysed projects. However, to be able to define how much is a “good” amount of AS in MAS platform is not a trivial task, because the answer is largely bounded to the development context (e.g., developers, developers skills, MAS platforms peculiarity). That is why in this study we mainly focused on the evolution of ADI and in grasping some insights about why they appear/disappear.

We can provide the answer to the first RQ:

RQ1: Which is the most present type of AS in MultiAgents Systems platforms? The most present type of AS (on average) is CD. The less present AS is HL.

We also ran the Mann-Kendall tests to analyse the trend of the same variables (CD, HL, UD, AS and ADI). Table 3 reports the results only of the significant cases, i.e., with p-value < 0.05 . The table also indicates whether the *trend* is increasing (+) or decreasing (-).

We now put in relation the results of the two analysis and provide a brief discussion of the architectural debt of each project. In particular, we manually checked the commit comments of each project, with a focus on the commits which presented large drops of the ADI value (points of interest). Our aim was to find a relationship between the change in the value of ADI and the content of the commit under analysis, starting from the description reported in the commit comment by the developers. For instance, if a sudden decrease in the ADI is backed by a comment stating that a major refactoring was applied in the commit, then the Arcan result is validated and we obtain an insight about practices for the removal of ATD.

Figure 1 depicts the ADI trend (y-axis) of the projects, computed for each commit (x-axis). Table 4 reports the main points of interest in the projects commit history, identified by the *Date* of the commit, the *Commit hash*, the *ADI* value and the interesting *Characteristics* of the commit. The table does not report results concerning Jade because we conducted a different

Table 2

Distribution analysis results

Metric	Jade	Jadex	Jason	Netlogo
CD mean	846.83	123.38	61.95	9.06
CD std.dev	7.96	55.78	13.22	4.31
CD min	837	1	48	1
CD max	856	193	91	16
HL mean	5	1.62	3.34	1.00
HL std.dev	0	0.49	0.58	NA
HL min	5	1	2	1
HL max	5	2	4	1
UD mean	28	67.96	8.55	1.79
UD std.dev	1.67	28.19	2.36	0.43
UD min	27	1	7	1
UD max	31	103	15	2
AS mean	879.83	192.12	73.84	7.2
AS std.dev	7.22	83.32	13.71	6.09
AS min	869	2	59	1
AS max	888	289	110	18
ADI mean	38.33	10.80	23.45	3.96
ADI std. Dev	1.97	5.95	3.06	3.52
ADI min	35	3	19	0
ADI max	41	23	30	11

Table 3

Mann - Kendall test results

Project	P-value	Variable	Trend
Jade	0.019	ADI	-
Jadex	0.000	ADI	+
Jadex	0.000	AS	+
Jadex	0.000	CD	+
Jadex	0.043	HL	+
Jadex	0.043	UD	+
Jason	0.003	AS	+
Jason	0.000	CD	+
Netlogo	0.000	ADI	-
Netlogo	0.000	AS	-

kind of analysis on it. Given that Jade is not hosted on Github, we could not analyse the commit comments, however we manually checked its changelogs.

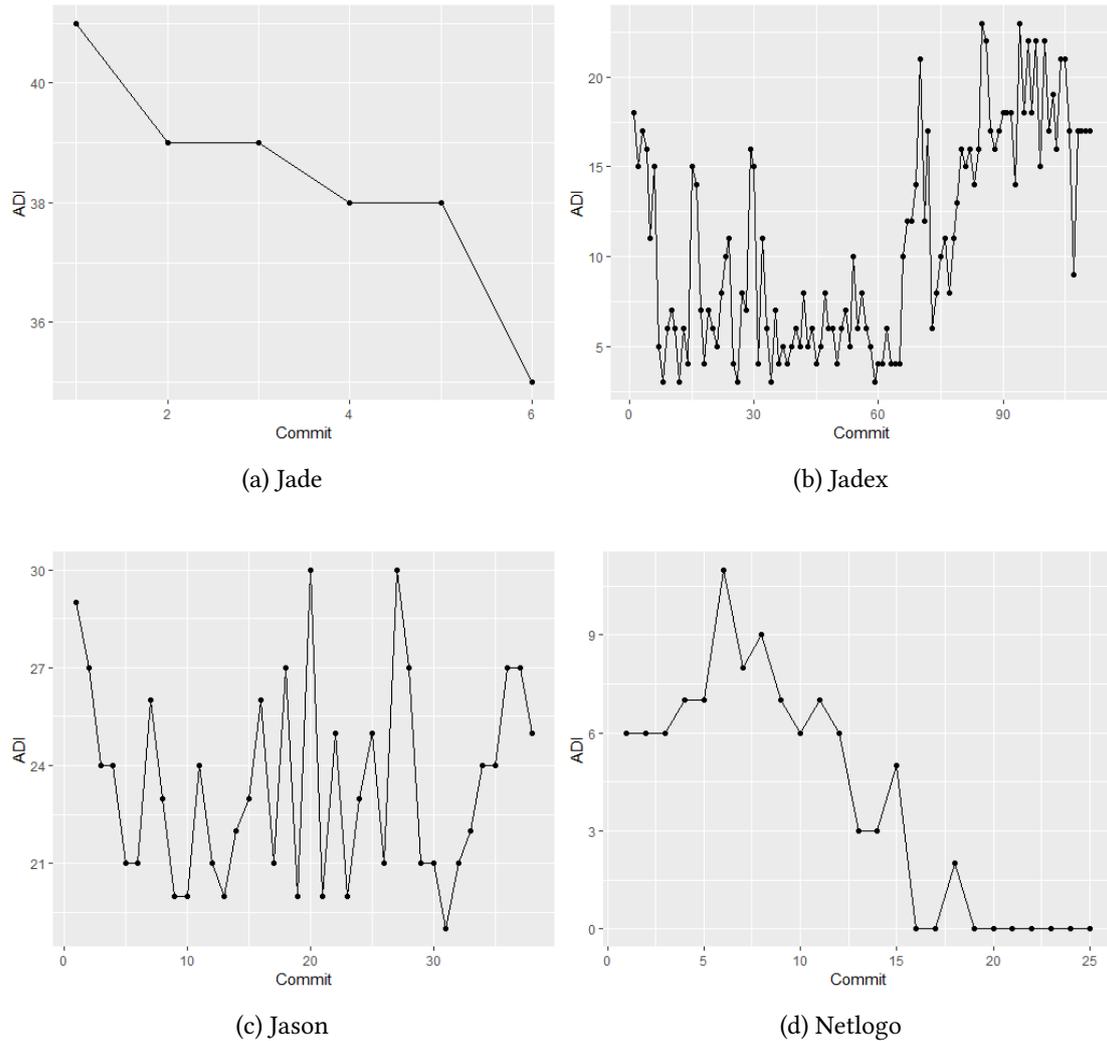


Figure 1: Evolution of ADI value of the 4 projects

4.1. Jade

As underlined before, the scarce number of analysed versions may have hindered the trend analysis results. However, the Mann-Kendall test gave an output for the ADI variable. In particular, the ADI trend is decreasing, but not dramatically. The detected ADI value ranges from 35 (last analysed version, 4.5.0) to 41 (first analysed version, 4.3.0). Given the few versions, we were able to manually analyse the changelog³ of all of them. We checked for key-terms, namely *Improvement(s)* and *Fix(es)*. We noticed that each version is characterised by many fixes, with version 4.4.0 having the greatest number of changelog comments addressing them (8). Concerning improvements, we identified few of them (approximately one per version), with

³<https://jade.tilab.com/doc/ChangeLog>

Table 4

List of ADI points of interest

Project	Date	Commit hash	ADI	Characteristics
Jadex	08/06/2009	09681d4371f53a1822de0a16c5b86e8349ea43c1	3	Min ADI value
Jadex	08/10/2009	5d266d02e4e9d90bde2dd942a7aff456eeca1aa4	3	Min ADI value
Jadex	14/12/2010	81be90b42d44a135e74d8a00213406951b78acaa	3	Min ADI value
Jadex	15/08/2011	a5b92b3f35a33222cd501bb37a2200e67d24187a	3	Min ADI value
Jadex	09/09/2014	fc28f548505583bfb2f1adcb1d3e368ec81aafd0	21	ADI drop, preceded by fixes and introduction of new data structure
Jadex	13/11/2017	6292d0cc21c24fa16627725e1bd0bfab52531222	9	ADI drop, preceded by fixes
Jason	20/04/2021	680921bbe8ff0247427d22e57ea3e36497143cd5	25	ADI drop, preceded by the implementation of a new test framework
Netlogo	07/02/2012	15daf0d82f11acc3a66ac9ca369fccf4efe77776	8	ADI drop
Netlogo	08/05/2012	5c5f707059b9a5c6546702cd2092b58e971b2632	6	ADI drop
Netlogo	17/05/2013	00ab7fae6c16c7a9b7e6b928080b15e0cd532e29	3	ADI drop
Netlogo	17/06/2013	82673cd6eda0f19b1dd533bd0e3a629ea24f23e6	3	ADI drop
Netlogo	31/01/2014	05710fe041397fd70286bc2347343993dd4f5563	0	ADI drop, corresponding to pull-request
Netlogo	13/03/2014	9056a8d98b69a2a984580d2cafceffc50685acb6	0	ADI drop, corresponding to pull-request
Netlogo	15/05/2014	c6a5902697212fb7adc14ae8d1e8a3e428e4dae8	2	ADI drop, corresponding to the addition of a new Scala submodule
Netlogo	04/09/2014	895609613fc1b5f592ff9eb84dcc5767f40ee7ec	0	ADI drop, corresponding to pull-request

most of them referring to enhancements to security. However, version 4.5.0 reports a comment about “Improved code style and logging”. A clean code style can improve maintainability, and this could be reason behind the ADI value of this version, the lowest detected.

4.2. Jadex

The ADI trend is increasing. The same happens for all the other variables (number of AS, CD, HL and UD). Indeed, Jadex is the project with the highest mean number of AS.

We manually analysed the points in time where ADI reached its lowest values, with the aim to understand whether interesting practices to manage architectural debt could emerge. In particular, we analysed the five commits corresponding to the lowest values of ADI, equals to 3 (see Table 4). Unfortunately, there are no messages or comments associated to those specific commits. The only interesting aspect is that all the five commits were created by the same two authors. We also analyse the period of time between and 09/09/2014 comments report multiple time the word “fix” and also the adoption of a dedicated info structure for Non-Functional

properties (NFPropertyInfo class). which are Non-functional property annotation.

Another point of interest in the Jadex history is on date 13/11/2017, when ADI drops from value 17 to 9. We checked the commit comments between the two points, corresponding to the changes made in a month, and all of them concern fixes. Some examples: “**Fix** proxy factory class loader issue and component spec as class.”; “**Fixed** most test failures caused by “config cleanup” commits”; “**Fix** component/bootstrap factory stored as string and as class”.

4.3. Jason

This projects ADI does not show any trend. However, its number of AS and CD has an increasing trend. We manually analysed a sampled period, which comprises the commits between 17/08/2020 and 20/04/2021. First we analysed the period from the high peak (ADI=30) to the lowest point (ADI=19). From the commit comments and the changelog of the nearest release, it appears that the most meaningful development was the implementation of a new tests framework. Even if it is affected by less AS with respect to Jadex, Jason is the project with the highest average ADI value. This means that compared to Jadex, its AS are more critical (have highest severity [4]).

4.4. NetLogo

NetLogo ADI trend is the only one decreasing. At the same time, the number of AS has a decreasing trend: we can deduce that the decrease of the value of ADI is not due to the decreased severity of the smells, but only due to the decrease of the total number of smells. In general, Netlogo is the projects less affected by architectural smells and with the lowest values of ADI (see Table 2).

We manually analysed the commit history of this project, in particular we focus on the points where ADI decreases (see Table 4). Most of the associated commit messages indicate improvements: “**Minor improvement** to Client Perspective Example.”; “**Mostly-irrelevant correction** to a HubNet method’s Scala style”. However, there are no signs of big, structural changes which could explain the significant drops of the ADI value, apart from the presence of three pull-requests, corresponding to $ADI = 0$ and the introduction of a new Scala submodule providing network analysis tools for use in NetLogo (commit message: “Add new network extension submodule!”).

RQ2: What can we observe according to architectural debt of the considered MultiAgents Systems platforms? All the analysed projects present architectural debt along their development history, but with different trends. Jadex has an increasing ADI, while Jade and Netlogo show a descreasing trend, with Netlogo having the last commits with zero debt. Jason did not present any trend.

5. Conclusions

We exploited our tool Arcan to analyse four Open-Source MultiAgent Systems (MAS) development platforms and we evaluated their Architectural Technical Debt (ATD). We investigated the

outcome of the tool by manually analysing the commit comments available on Github, for three of the four projects, and the changelogs for one of them. From our analysis, we acknowledged that the considered MAS platforms are affected by architectural debt, in particular Jade is the most affected, while Netlogo is the less affected, with a decreasing ADI trend.

From the manual analysis, we could not find clear indication of practices to manage architectural debt. However, for all the projects, in the points in time where ADI reaches its minimum, the comments refer to “Bug fixing”, “Improvements” or pull requests. This could mean that architectural debt, usually considered only at architectural level, has also a relationship with issues at code level, such as bugs.

Our findings suggest us possible future works. As just outlined, studying the correlation between MAS platform architectural debt and bugs could lead to the conclusion that code level bugs have an impact on the accumulation/decrease of ATD. Moreover, the validation of the ATD values found in the projects could be refined by testing the correlation with issues coming from issue trackers (e.g. Jira⁴). In this way, we could further investigate whether “Improvements” (which is usually recognized as a category of issues) do have a relationship with the decrease of ATD. Another interesting study could investigate the relationship between ATD and MAS platforms’ performance, since a link between performance and design decisions has already been proven (see Section 2).

Another next natural step will be to apply this kind of analysis to real MASs developed with these platforms, or to enlarge our analysis to the many add-ons of these four platforms (for example WSIG and OntologyBeanGenerator for Jade [26, 27, 28]), to study if we can identify some common ATDs for MASs or further problems in MAS development platforms. We have a long experience in developing large and real MASs ([29, 30, 31]), and we would search for other concrete examples of large MASs to be analyzed from this architectural point of view.

References

- [1] Z. Li, P. Liang, P. Avgeriou, Architectural debt management in value-oriented architecting, in: *Economics-Driven Software Architecture*, Elsevier, 2014, pp. 183–204.
- [2] N. A. Ernst, S. Bellomo, I. Ozkaya, R. L. Nord, I. Gorton, Measure it? manage it? ignore it? software practitioners and technical debt, in: *Proc. of the 2015 10th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2015*, ACM, New York, NY, USA, 2015, pp. 50–60.
- [3] W. Cunningham, The wycash portfolio management system, *ACM SIGPLAN OOPS Messenger* 4 (1992) 29–30.
- [4] R. Roveda, F. Arcelli Fontana, I. Pigazzini, M. Zanoni, Towards an architectural debt index, in: *Proceedings of the Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Technical Debt track*, IEEE, Prague, Czech Republic, 2018.
- [5] G. Digkas, M. Lungu, A. Chatzigeorgiou, P. Avgeriou, The evolution of technical debt in the apache ecosystem, in: *European Conference on Software Architecture*, Springer, 2017, pp. 51–66.

⁴<https://www.atlassian.com/software/jira>

- [6] S. S. de Toledo, A. Martini, A. Przybyszewska, D. I. Sjøberg, Architectural technical debt in microservices: a case study in a large company, in: 2019 IEEE/ACM International Conference on Technical Debt (TechDebt), IEEE, 2019, pp. 78–87.
- [7] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, D. Dennison, Hidden technical debt in machine learning systems, *Advances in neural information processing systems* 28 (2015) 2503–2511.
- [8] I. Pigazzini, F. Arcelli Fontana, Evaluating the architectural debt of iot projects, in: 3rd International Workshop on Software Engineering Research & Practices for the Internet of Things (SERP4IoT 2021), 2021.
- [9] R. C. Cardoso, A. Ferrando, D. Briola, C. Menghi, T. Ahlbrecht, Agents and robots for reliable engineered autonomy: A perspective from the organisers of AREA 2020, *J. Sens. Actuator Networks* 10 (2021) 33. URL: <https://doi.org/10.3390/jsan10020033>. doi:10.3390/jsan10020033.
- [10] V. Mascardi, D. Weyns, A. Ricci, Engineering multi-agent systems: State of affairs and the road ahead, *ACM SIGSOFT Softw. Eng. Notes* 44 (2019) 18–28. URL: <https://doi.org/10.1145/3310013.3310035>. doi:10.1145/3310013.3310035.
- [11] F. Zambonelli, A. Omicini, Challenges and research directions in agent-oriented software engineering, *Auton. Agents Multi Agent Syst.* 9 (2004) 253–283. URL: <https://doi.org/10.1023/B:AGNT.0000038028.66672.1e>. doi:10.1023/B:AGNT.0000038028.66672.1e.
- [12] R. Calegari, G. Ciatto, V. Mascardi, A. Omicini, Logic-Based Technologies for Multi-Agent Systems: Summary of a Systematic Literature Review, *International Foundation for Autonomous Agents and Multiagent Systems*, Richland, SC, 2021, p. 1721–1723.
- [13] F. Arcelli Fontana, I. Pigazzini, R. Roveda, D. A. Tamburri, M. Zanoni, E. D. Nitto, Arcan: A tool for architectural smells detection, in: *Int'l Conf. Software Architecture (ICSA 2017) Workshops*, Gothenburg, Sweden, 2017, pp. 282–285. doi:10.1109/ICSAW.2017.16.
- [14] O. F. Rana, K. Stout, What is scalability in multi-agent systems?, in: *Proceedings of the fourth international conference on Autonomous agents*, 2000, pp. 56–63.
- [15] R. Endsuleit, J. Calmet, et al., A security analysis on jade (-s) v. 3.2, in: *Proceedings of NORDSEC*, Citeseer, 2005, pp. 20–28.
- [16] L. Mulet, J. M. Such, J. M. Alberola, Performance evaluation of open-source multiagent platforms, in: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 2006, pp. 1107–1109.
- [17] J. M. Alberola, J. M. Such, A. Garcia-Fornes, A. Espinosa, V. Botti, A performance evaluation of three multiagent platforms, *Artificial Intelligence Review* 34 (2010) 145–176.
- [18] F. A. Fontana, F. Locatelli, I. Pigazzini, P. Mereghetti, An architectural smell evaluation in an industrial context, in: *The Fifteenth International Conference on Software Engineering Advances, ICSEA 2020*, 18-22 October 2020, Porto, Portugal, 2020.
- [19] F. L. Bellifemine, G. Caire, D. Greenwood, *Developing Multi-Agent Systems with JADE*, Wiley, 2007.
- [20] L. Braubach, W. Lamersdorf, A. Pokahr, Jadex: implementing a bdi-infrastructure for jade agents, *EXP In Search of Innovation (Special Issue on JADE)* 3 (2003).
- [21] R. H. Bordini, J. F. Hübner, M. Wooldridge, *Programming Multi-Agent Systems in AgentSpeak Using Jason (Wiley Series in Agent Technology)*, John Wiley & Sons, Inc., USA, 2007.

- [22] U. Wilensky, NetLogo, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999. URL: <http://ccl.northwestern.edu/netlogo/>.
- [23] N. Nagappan, T. Ball, Using software dependencies and churn metrics to predict field failures: An empirical case study, First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007), 2007, pp. 364–373.
- [24] E. Kouroshfar, M. Mirakhorli, H. Bagheri, L. Xiao, S. Malek, Y. Cai, A study on the role of software architecture in the evolution and quality of software, volume 2015-Augus, IEEE International Working Conference on Mining Software Repositories, IEEE Computer Society, 2015, pp. 246–257. doi:10.1109/MSR.2015.30.
- [25] R. C. Martin, Object oriented design quality metrics: An analysis of dependencies, ROAD 2 (1995).
- [26] D. Briola, V. Mascardi, M. Gioseffi, Ontologybeangenerator 5.0: Extending ontology concepts with methods and exceptions, in: M. Cossentino, L. Sabatucci, V. Seidita (Eds.), Proceedings of the 19th Workshop "From Objects to Agents", Palermo, Italy, June 28-29, 2018, volume 2215 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2018, pp. 116–123. URL: http://ceur-ws.org/Vol-2215/paper_19.pdf.
- [27] C. van Aart, R. Pels, G. Caire, F. Bergenti, Creating and using ontologies in agent communication, in: Proc. of OAS, 2002.
- [28] M. Tomaiuolo, F. Bergenti, A. Poggi, P. Turci, Owlbeans - from ontologies to java classes, in: M. Baldoni, F. D. Paoli, A. Martelli, A. Omicini (Eds.), WOA 2004: Dagli Oggetti agli Agenti. 5th AI*IA/TABOO Joint Workshop "From Objects to Agents": Complex Systems and Rational Agents, 30 November - 1 December 2004, Torino, Italy, Pitagora Editrice Bologna, 2004, pp. 116–125.
- [29] D. Briola, V. Deufemia, V. Mascardi, L. Paolino, Agent-oriented and ontology-driven digital libraries: the indianamas experience, *Softw. Pract. Exp.* 47 (2017) 1773–1799. URL: <https://doi.org/10.1002/spe.2494>. doi:10.1002/spe.2494.
- [30] M. Leotta, S. Beux, V. Mascardi, D. Briola, My mood, a multimedia and multilingual ontology driven MAS: design and first experiments in the sentiment analysis domain, in: C. Bosco, E. Cambria, R. Damiano, V. Patti, P. Rosso (Eds.), Proceedings of the 2nd International Workshop on Emotion and Sentiment in Social and Expressive Media: Opportunities and Challenges for Emotion-aware Multiagent Systems co-located with 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Istanbul, Turkey, May 5, 2015, volume 1351 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2015, pp. 51–66. URL: <http://ceur-ws.org/Vol-1351/paper4.pdf>.
- [31] V. Mascardi, D. Briola, D. Ancona, On the expressiveness of attribute global types: The formalization of a real multiagent system protocol, in: M. Baldoni, C. Baroglio, G. Boella, R. Micalizio (Eds.), AI*IA 2013: Advances in Artificial Intelligence - XIIIth International Conference of the Italian Association for Artificial Intelligence, Turin, Italy, December 4-6, 2013. Proceedings, volume 8249 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 300–311. URL: https://doi.org/10.1007/978-3-319-03524-6_26. doi:10.1007/978-3-319-03524-6_26.

Prototypes of Productivity Tools for the Jadescript Programming Language

Giuseppe Petrosino¹, Eleonora Iotti¹, Stefania Monica² and Federico Bergenti¹

¹*Dipartimento di Scienze Matematiche, Fisiche e Informatiche, Università degli Studi di Parma, Italy*

²*Dipartimento di Scienze e Metodi dell'Ingegneria, Università degli Studi di Modena e Reggio Emilia, Italy*

Abstract

Jadescript is an agent-oriented programming language built on top of JADE. So far, the focus of the development of the language was on design choices, on syntax refinements, and on the introduction of expressions and constructs for agent-related abstractions and tasks. In this paper, a proposal to achieve the crucial goal of making Jadescript suitable for professional use is presented. The success of Jadescript, as a solid language to build real-world agent-based software systems, is necessarily related to its effective integration with mainstream development tools. In this paper, some of the productivity tools developed to integrate Jadescript with a mainstream development environment are presented as a way to promote the successful adoption of the language towards the community of JADE users.

Keywords

Agent-Oriented Programming Languages, Agent-Oriented Software Engineering, Jadescript, JADE

1. Introduction

The search for novel and effective development technologies to design agents and to build multi-agent systems is a fast-growing research issue. Over the years, agents were employed in many different application scenarios, as illustrated, e.g., in [1], where the relevant impact of agent technologies is discussed. Example application scenarios are agent-based simulations [2], distributed constraints reasoning [3], accurate indoor positioning [4, 5, 6, 7, 8, 9, 10], serious games [11, 12, 13], and network management [14], just to cite some.

In general, multi-agent systems are key tools for those problems when goals to be achieved are sufficiently complex to require the coordination and cooperation of a large number of different agents, often distributed on many hosts, that must use their skills in an effective way to achieve a collective goals. Multi-agent systems open to a plethora of new design and development problems, such as agent communication and interaction protocols, message passing and routing, deployment of agents to network hosts, the reception of shared environment information, norms and validations, and so on. Some of such interesting problems are taken over by *AOSE (Agent Oriented Software Engineering)* [15] researchers to produce a remarkable range of solutions and frameworks, as well as sophisticated software platforms [16].

WOA 2021: Workshop "From Objects to Agents", September 1–3, 2021, Bologna, Italy

✉ giuseppe.petrosino@studenti.unipr.it (G. Petrosino); eleonora.iotti@unipr.it (E. Iotti);

stefania.monica@unimore.it (S. Monica); federico.bergenti@unipr.it (F. Bergenti)

🆔 0000-0001-6234-5328 (G. Petrosino); 0000-0001-7670-2226 (E. Iotti); 0000-0001-6254-4765 (S. Monica);

0000-0002-4756-4765 (F. Bergenti)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Most, if not all, AOSE methods and tools target the *AOP (Agent-Oriented Programming)* [17, 18] paradigm, which explicitly treat the concept of agent as a basic building block, ready to be used by the programmer. Nonetheless, many AOP frameworks and platforms provide software libraries written in some *GPL (General Purpose Language)*, and such libraries extend the usage of the chosen language to AOP problems. As a matter of fact, a common way to enrich a GPL with custom functionality, e.g. agent-based functionality, is to provide specific *APIs (Application Programming Interfaces)*. Alternatives to the use of GPLs in software development are various, and its worth citing at least two of such possibilities, namely, *DSLs (Domain Specific Languages)* and scripting languages.

These two alternatives, revisited in AOSE, open to a relevant advancement in the direction of AOP, which is the use of *APLs (Agent Programming Languages)* [19]. Such languages not only support agent-based features, but they also put them on a language level. Jadescript [20, 21, 22], a language built on top of *JADE (Java Agent DEvelopment framework)* [23, 24], is an example of an AOP scripting language. Jadescript is a significant extension of a previous APL called *JADEL* [25, 26, 27, 28, 29, 30, 31] that incorporates the features of an AOP scripting language. Other popular APLs are *Jason* [32], which is an implementation of *AgentSpeak(L)* [33], *3APL* [34], *GOAL* [35], *SARL* [36], and several others [37].

It is common opinion that the major benefit of adopting a specific language for agent development is the availability of native abstractions, constructs, and expressions in the language to explicitly recall the agent domain, putting agents as first-class citizens of the language. On the other hand, despite the great results in terms of effectiveness and usability, most of the aforementioned APLs has a niche user base, composed mainly by researcher and students. Pure agent-based programming seems yet relegated to academic environments, despite many real-world applications use multi-agent systems on a daily basis. A reason for that, among many others, lies in the preferences of programmers and in programming trends. A successful language for a wider audience must take into account such preferences and trends, making its idioms as simple and readable as possible, yet not ambiguous. Unfortunately, these design choices alone are not sufficient to bring success to programming languages. The core functionalities offered by a language are appreciated when they are stable and reliable, making the language usable for robust applications. Therefore, the proposal of a novel APL should not only regard the accurate design and implementation of desired functionality by means of syntactical categories and their semantics, but also the construction of an adequate ecosystem. Such an ecosystem could be defined as the set of all tools, utilities, and interfaces that help programmers in their daily coding routine, i.e., those services offered by *IDEs (Integrated Development Environments)*, frameworks, libraries, and related tools.

In this paper, the main steps taken to build such tools, utilities, and interfaces for the Jadescript language are described. The adopted approach aims at making the language completely integrated with the Eclipse IDE [38] and its plugins, thus providing tools such as a Jadescript perspective, some specific Eclipse wizards, a dedicated syntax highlighting editor, and a launch system for agents and agent containers. This work is primarily based on another Eclipse plugin, called Xtext, which generates some ready and easy-to-use tools for DSLs. Such tools were then adapted to the agent domain and made suitable for the user experience envisaged for Jadescript programmers. The resulting ecosystem is an important step for the growth of Jadescript, and it brings to the language the professional feeling that JADE users expect.

This paper is structured as follows. First, in Section 2, an introduction to Xtext and related technologies is provided to give the reader sufficient background information on how to build a professional tool for the Eclipse IDE using Xtext. Then, in Section 3, the core features of the Eclipse plugin for Jadescript are detailed, taking into account the Xtext extensions and the Eclipse extensions. Finally, a discussion on the main results of the approach adopted in the development of presented tools is provided to conclude the paper.

2. Overview of Xtext

Xtext [39] is the main software used to create the presented tools related to Jadescript. Xtext is an open-source framework for the development of DSLs and programming languages. It is designed to lift most of the burden of the programming language designer, not only by taking the usual tasks of a parser generator, but also by providing a set of advanced tools that guide in the construction of a complete compiler and a full-featured IDE.

An Xtext language project is made of several related Eclipse projects. Three of them are the most important:

1. The main project, which contains the grammar, the support for the semantics, and all the other components for the language that are independent from the *UI (User Interface)*;
2. The IDE project, which contains the code for the general behaviour of the UI, regardless of the specific target environment, so the code in this project can be specialised, for example, for an IntelliJ IDEA [40] plugin, or for an editor embedded in a Web page; and
3. The UI project, which depends on the IDE project and contains the specific details related to the Eclipse UI to implement a custom language plugin for the Eclipse IDE.

The main project contains the entry point for the language design process, which is the Xtext grammar file for the language. Xtext is grammar driven and it provides a grammar language to generate, from a single source file, all the essential elements of the skeleton of the compiler and of the other tools.

The first essential component generated by Xtext is the lexer, which is generated from the terminals defined the grammar. The lexer is usually complete and sufficient in most of the cases. However, it is worth mentioning that the default behaviour of the lexer was specialised for Jadescript because Jadescript is a language with semantically relevant indentation. Section 3 describes the details of this specialisation.

The second component generated by Xtext is the parser. The parser is used by the compiler and by the editor to obtain an *AST (Abstract Syntax Tree)* from a processed text. Since the Xtext grammar language is an extension of the grammar language of the ANTLR [41] parser generator, the generated parser employs a *LL(*)* [42] parsing strategy.

Together with the parser, Xtext generates a syntax validator. This component is in charge of isolating the portions of a processed text that are syntactically incorrect to produce the corresponding errors, warnings, and recommendations for the user.

Finally, Xtext produces a metamodel of the generated language using *EMF (Eclipse Modeling Framework)* and its metamodel format, called Ecore. A set of Java interfaces and classes is created to represent an object model of the grammar. For each non-terminal grammar rule, a

Java `EObject` class is generated, where each component of the rule is mapped to one of the properties of the class. This provides the language designer with a statically typed programming interface to work with the ASTs generated by the parser. Note that an important difference between an Xtext grammar and an ANTLR grammar is the possibility to add additional metadata to customise the aspects of the generation of the object model of the language. Another notable difference between the two grammar languages is the possibility to inject, directly in the model of the generated AST, useful metadata like, for example, the type of syntax element that a reference can be linked to. Such a metadata comes in handy when working with the AST in the portions of the compiler that define the semantics of the language.

With these essential components automatically generated by Xtext from the grammar file, the language designer has already a working editor and other working tools for code editing and syntactical validation. However, to actually interpret and generate executable code, it is required to add custom components. This is achieved by a widespread adoption of the *DI (Dependency Injection)* design pattern. The main idea of such an approach is that all the classes that implement the functionalities provided by the final plugin refer to their main dependencies by declaring them as fields annotated with a dedicated annotation. The Xtext framework uses a class to define the bindings among the Java interfaces of these dependencies with their respective implementations. Such bindings are then used by an injector object to create all the components and their dependencies at runtime. The module class is open for extensions and all the binding methods can be overridden to provide the language designer with the ability to change, in a controlled and structured way, almost any aspect the generated tools.

Among such customisable aspects, two require particular attention, namely code validation and code generation. Semantic validation of the code is managed by a validator class. In Xtext, such a class is implemented with a declarative approach. The language designer specifies, by adding methods to the class, which types of the Java object model of the AST are to be checked. In case of erroneous or problematic code, such methods can build and report errors, warnings, and recommendations that are used as feedback to the user.

Code generation can be achieved by means of a class that extends the `IGenerator` interface, which defines how to generate new texts starting from the AST obtained from the parsing of a source file. For languages targeting the *JVM (Java Virtual Machine)*, however, another specialised approach is available, based on the `JVMModelInferer` class. By extending this class, the language designer is able to symbolically declare which Java classes, interfaces, methods, and fields are generated from each source file. Xtext keeps track of such mappings and use them to ease the implementation of the language tools by partially generating:

1. A type checking system, based on the Java type system;
2. A scope provider, which is able to resolve references to Java packages, types, and symbols;
3. A code generator, which creates the Java source files corresponding to the declared Java structures provided by the `JVMModelInferer` object; and
4. Some IDE features like basic auto-completion support, symbolical navigation in the editor, linking between written and generated code, and basic refactoring.

Obviously, if the validator or the type checker find errors in the source code, the compiler aborts code generation, signalling the problems to the user.

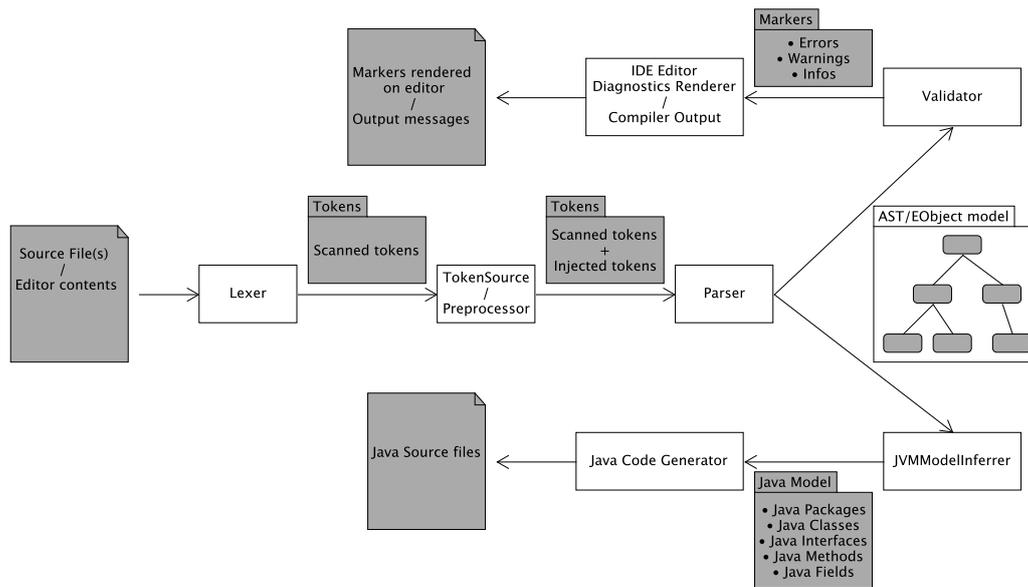


Figure 1: The compilation and validation processes in the Xtext framework for languages that target the Java virtual machine.

Jadescription is currently a language based on the JVM. More specifically, it is compiled to Java code. For this reason, the Jadescription compiler uses the `JVMMoelInferer` class to take advantage of the prebuilt mechanisms generated by Xtext for JVM-based languages. The general outline of the validation and code generation processes provided by Xtext and used by the Jadescription compiler is schematised in Fig. 1.

When the user feeds the compiler with a set of Jadescription source files, the Xtext runtime provides the contents of the files to the lexer generated from the Jadescription grammar. The lexer produces a stream of tokens, and the `TokenSource` interface, provided by Xtext, is used to preprocess the stream of tokens to inject into the stream needed synthesised tokens relative to the semantically relevant indentation. Actually, the original stream of tokens is analysed to identify the points in the stream where the level of indentation changes. Synthesised tokens are injected at the identified points in the stream to ensure that changes in the indentation are properly reported to the parser. The new stream of tokens is fed to the parser, which produces an AST and an EMF model of it. These results are then fed to the validator, which statically analyses the code in search for problems, following the semantic rules of the language. If the validator does not find any errors in the code, the same AST is reused and provided to the `JVMMoelInferer` class, which produces an intermediate representation, namely the Java model of the target code. This representation is finally used by the framework to produce the Java code that Eclipse eventually compiles for the JVM.

3. The Jadescript Eclipse IDE Plugin

The discussed Jadescript development tools include an Eclipse IDE plugin, which contains all the software tools to write Jadescript code, to create and manage projects, to create and edit source files, and to launch and debug agents. The plugin was created with the help of Xtext, especially for those features of it that are strongly related with the syntax and the semantics of the language, and that, therefore, are part of the Jadescript compiler. However, some of the tools, as discussed in this section, were created by means of the tools provided by the Eclipse PDE (*Plugin Development Environment*).

3.1. Xtext Extensions

As mentioned in Section 2, in Xtext, new language projects enjoy of a set of interesting features implemented by default and generated automatically from the grammar file. However, some aspects of the generated code require the language developer to adapt the default Xtext implementations by overriding specific methods with custom methods. For Jadescript, two aspects required specialised implementations, namely, the `TokenSource` implementation used to manage semantically relevant indentation, and the set of mechanisms that implement the semantics of the language in the compiler.

3.1.1. Token Source System

In an Xtext-generated parser, the parser reads the input tokens from a `TokenSource` object, which is an object that produces tokens from source code upon request from the parser. In most programming languages, the parser assumes that whitespace characters (i.e., blanks, tabulations, newline characters) are hidden in the grammar and not considered in the input stream of tokens. The purpose of whitespace characters is to act as separators among parts of the text that are relevant for the grammar. This is not completely true for languages with semantically relevant indentation like Jadescript. In such languages, the level of indentation of a line not only keeps the code tidy and easy to read, but it is also used by the compiler to understand where the line is placed in the structure of the code. For example, for procedural code, some statements are expected to include inner blocks of code (e.g., the *then* branch of an *if* statement, or the body of a *loop* statement), and the lines belonging to such blocks of code have to start with an inner level of indentation. At the same time, in Jadescript and other modern programming languages, the sequential composition of statements is not expressed by an explicit *end-of-statement* operator symbol (e.g., `;` for C-like languages). Such separation of statements has to be inferred by the compiler using the newline character as hint. In this inference mechanism, the compiler has to leave the possibility for the user to split any statement in two or more lines, whenever such statements are too long and the programmer wishes to increase readability.

In Jadescript, this set of behaviours is encoded in a special kind of tokens (also known as synthetic tokens) that signals the parser of three types of relevant points in the code:

1. The point of termination of a statement (*NEWLINE* synthetic token);
2. The point where a new code block is opened (*INDENT* synthetic token); and
3. The point where a previously opened code block is closed (*DEDENT* synthetic token).

The `TokenSource` interface of Xtext is then implemented by the `JadscriptTokenSource` class that includes an algorithm that injects the synthetic tokens mentioned above according to a simple set of rules.

By default, when the parser requests a new token, the `JadscriptTokenSource` object simply responds with the next token that the lexer generated by scanning the source code. However, when at least one newline characters is encountered in the stream, the `JadscriptTokenSource` object computes the indentation level of the new line. If the indentation is more in depth than the previous line, and the previous line ends with `do` (keyword that, in Jadscript, is used in many constructs to express the beginning of the definition of a procedural body) or the new line starts with any of the following keywords {`concept`, `proposition`, `predicate`, `action`, `function`, `procedure`, `on`, `property`, `execute`}, then an `INDENT` synthetic token is injected in the stream. When the indentation is more in depth than the previous line, but those keywords are not present, the `JadscriptTokenSource` object does not inject any new token in the stream. This last rule allows users to split a line into two, indenting the second line, to simply improve readability without changing the semantics of the code in the lines. If, however, the indentation level is the same as the previous line, a `NEWLINE` token is injected, signalling the parser that a statement (or declaration) ended with the previous line and that a new statement (or declaration) starts with the new line. Finally, when the indentation is less in depth than the previous line, a number of `DEDENT` tokens are injected corresponding to the number of blocks being closed.

3.1.2. Jadscript Semantic Classes

After parsing, the compiler created by the Xtext framework produces a Java object model of the AST. This can be navigated to perform the computations required by the semantics of the language. These compiler computations, in the Jadscript compiler, are handled by a set of Java classes called semantic classes. Each one of these classes handles how a particular node of the AST is used for code generation and validation.

The semantic classes can be subdivided in four categories, each one referring to a type of construct of the language. The following paragraphs describe these categories, sorted by structural depth level.

Top Level Entities. These semantic classes implement the semantics of those top-level declarations (e.g., *agent*, *behaviour*, and *ontology*) that can be written directly inside a file, not contained in any other construct or declaration. When the compiler walks the AST on these types of nodes, they are mapped directly to JVM types (classes and interfaces) by means of the utilities provided by the `JVMModelInferer` class generated by Xtext.

Entity Features. These elements of the language are the main building blocks of each top-level declaration. They are usually directly enumerated in the body of the declaration, and for this reason, they appear as indented by just one level in the source code. Examples of features include *event handlers* and *properties* in *agent* and *behaviour* declarations, and *concept* and *predicate* entries in *ontology* declarations. Entity features are usually compiled to Xtext-compatible JVM model elements, namely Java fields, methods and inner classes.

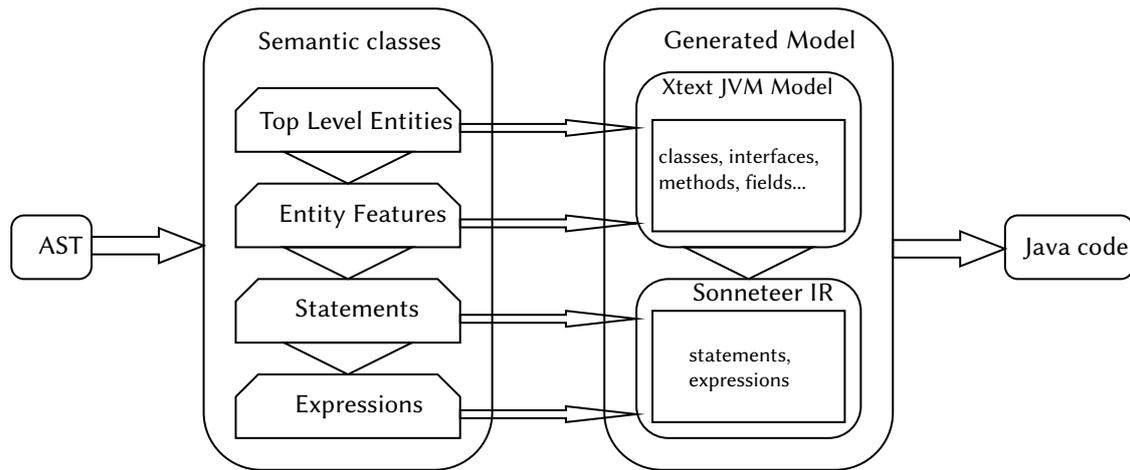


Figure 2: The categories of semantic classes and how their generated artefacts are composed.

Statements. Statements are the building blocks of the procedural portions of code. They are used in those entity features that require a procedural body, i.e., structured lists of commands. Note that such commands can include expressions or other procedural bodies (e.g., the guard and the body of a `while` statement). Statements are compiled by semantic classes into objects of a custom *IR (Internal Representation)* model, named Sonneteer. Sonneteer is a small Java library which implements a simple API to generate text strings of Java source code. The usage of this library in the implementation of the semantic classes ensured a good degree of type safety in the generation of structured Java code. Objects built with this library are used by the compiler in the code generation phase at the end of compilation, to compute the actual text content of the generated Java source code. The `send message` statement and the `activate behaviour` statement are good examples of elements of this category.

Expressions. As in many modern programming languages, the most fine-grained category of language constructs is expressions. Expressions are designed to be composable, and, in statically and strongly typed languages like Jadescript, each expression has a type, which is used by the compiler to check if some combinations of operations and operands is consistent with the rules of the language semantics. In Jadescript, the type of an expression is computed at compile time not only for validation purposes, but also to infer the type of variables and of agent/behaviour properties. Semantic classes related to this category compile expressions into simple text strings in three steps. In the first step, all the statements and expressions in a procedural code block are translated into Sonneteer objects, which include placeholder elements that annotate the generated code with compiler metadata. These placeholder elements are then analysed in the second step. The result of the analysis is finally used in the third step to perform optimisations. Note that expressions include literals, infix and unary operations like addition and logical negation, and other special operations like `matches` for pattern matching.

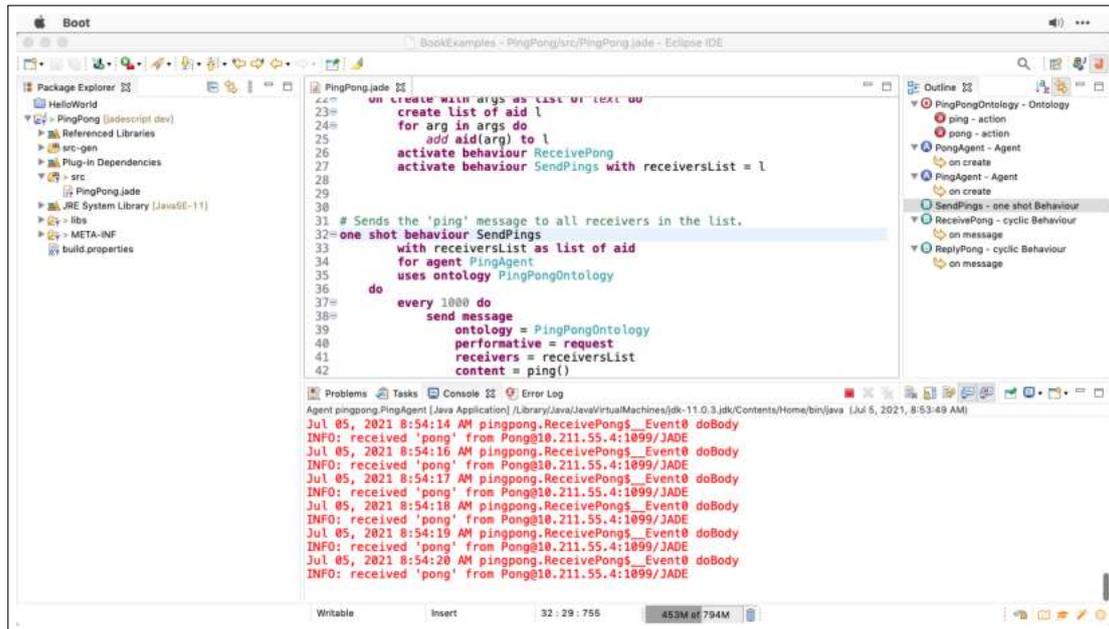


Figure 3: A screenshot of the Jadescript perspective hosted in the Eclipse IDE.

3.2. Eclipse Extensions

Part of the Jadescript Eclipse plugin is implemented using directly the PDE. The Eclipse IDE is designed as an extensible framework, with facilities that ease the addition and customisation of functionality. Such customisations can be created in plug-ins, and the extensibility approach allows plug-ins to use and extend other plug-ins declaring a set of hooks in the extension points in the manifest XML file of the plugin. The Jadescript plugin uses these extension points to customise some aspects of the user interface of the IDE. The main extensions of the Eclipse IDE provided by the plugin are the Jadescript Perspective, the Wizards, the Syntax Highlighting in the Jadescript editor, and the Container and Agent launcher actions.

3.2.1. Jadescript Perspective

Fig. 3 shows a screenshot of the Jadescript Perspective as provided by the plugin. On the left, there is the *Package Explorer* View. It is a tree view of the current eclipse workspace and the contents of its projects. The contextual menu on these elements contains a set of common Eclipse project management actions, like importing and exporting, and operations to manage the view itself, like *Refresh*. However, it is enriched of several actions specific for the management of Jadescript projects, namely actions to start wizards for the creation of Jadescript projects and files (see section 3.2.2) and for running the Jadescript agents declared in the source files (see section 3.2.4). At the centre, there is the editor section. This acts as a container for editor tabs, including instances of the Jadescript Editor for the editing of Jadescript source files. On the right, the Jadescript perspective lays out the *Outline* View. This view shows the structure of the

code of the currently focused file in a tree view. In this view the root nodes of the tree represent the top level declarations in the file, and their children represent their declared features, i.e., properties, event handlers, functions and procedures for agent and behaviour declarations, and concepts, predicates, propositions and actions for ontology declarations. The Outline View enjoys of a bidirectional linking between the contents of the file and the nodes, which is updated and rendered in real time.

3.2.2. Eclipse Wizards

The plugins includes a set of wizards for the creation of projects and source files. The *New Jadescript Project* wizard guides the user in the creation of a new project with the Jadescript nature. Jadescript projects always include three folders, created by the wizard. The `src` directory is where all Jadescript and Java source files written by the user should go. Jadescript files saved in this directory (and in its subdirectories) are used as input for the Jadescript compiler, which generates the corresponding Java files into the second directory, named `src-gen`. Finally, the `libs` directory contains a set of *JAR (Java ARchive)* libraries used by the project. The *New Jadescript Project* wizard always puts three JARs in this directory, which are the `jadescript.jar` file, which includes some required code for Jadescript (like the implementation of the base Jadescript *Agent* and *Behaviour* types), and the `jade.jar` and the Apache Commons Codec libraries, required for running JADE and Jadescript agents and platforms.

Four more wizards are used to guide the user to create new Jadescript source files. The *New Jadescript File* wizard creates a new empty Jadescript file in the specified project location, with the specified module. This wizard is specialised into the *New Jadescript Agent*, *New Jadescript Ontology* and *New Jadescript Behaviour* wizards, which collect information from the user in order to create new Jadescript source files with the stubs of, respectively, an agent declaration, an ontology declaration, and a behaviour declaration.

3.2.3. Syntax Highlighting

As many modern programming editors, code is highlighted with different colours, in order to help the user quickly recognise and tell the various elements of the code. This aspect of the appearance of code in the Jadescript editor can be customised by the user via the Jadescript section of the Eclipse preferences, at the *Syntax Coloring* preference page. The syntax highlighting for Jadescript is advanced enough to make use of complex semantic rules to highlight the text of different colours, and this is done by using a set of special methods in the semantic classes. This approach allows, for example, to highlight with different colours the first assignment of a variable and its subsequent usages and re-assignments.

3.2.4. Container and Agent Launchers

Two fundamental entries in the extension points of the plugin implement two actions in the IDE. The first is accessible from the Eclipse toolbar, and it is used to launch a new JADE main container. By pressing this button, a new instance of the JADE Main container is launched locally on the machine where Eclipse is running. This action creates a new local agent platform, and it can be used as a starting point to build a complex network of JADE containers. The

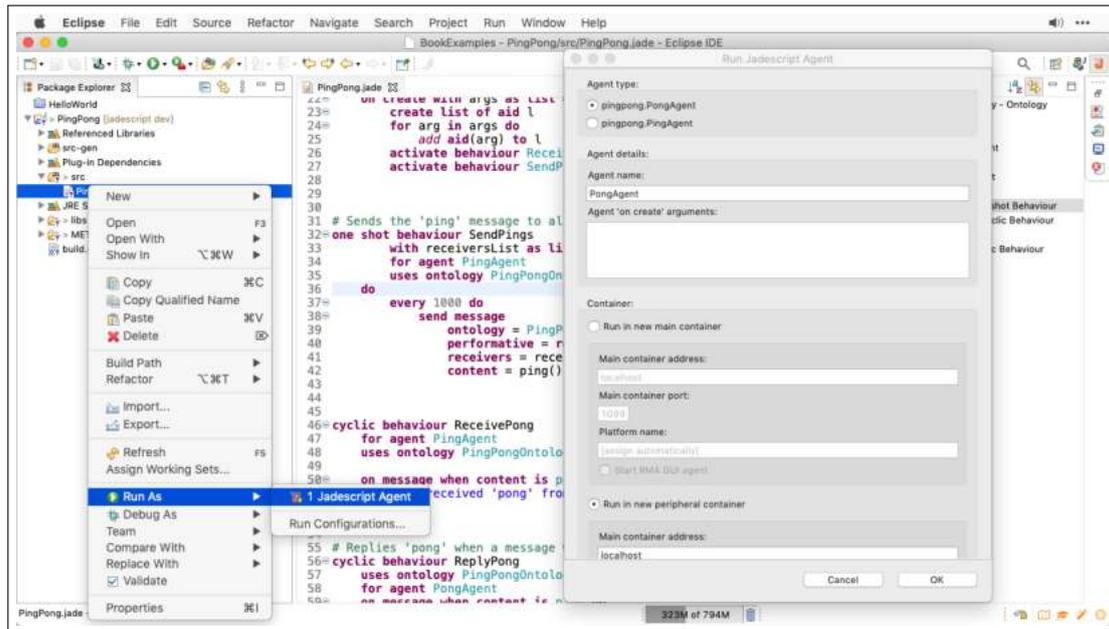


Figure 4: A screenshot of the agent launcher dialog for the Jadescript editor in the Eclipse IDE.

button is a pull-down button, with a second option available in its drop-down menu. By clicking on the second option, the created JADE main container includes a *RMA (Remote Monitoring Agent)* with a *GUI (Graphical User Interface)* that allows the developer to see the status of the platform and to create new containers and agents. The JADE software is launched using the Java classpath of the currently open project. In this way, from within the RMA GUI it is possible to launch new agents in the platform, using the Java classes agent generated from the Jadescript code.

The second action is accessible from the *Run As* submenu of the contextual menu. It is only accessible when a Jadescript source file is selected or open in the editor. When clicked, a dialog window opens. This window provides the user with the ability to launch a new agent in a new container. It itemises a list of selectable agent types, which correspond to the ones declared in the source file. The dialog then allows the user to customise various details of the agent, like its name, its input arguments, and the details of the container in which it will be created in.

4. Conclusions

In this paper, some of the productivity tools explicitly designed for Jadescript are presented and discussed. The approach that underlies such tools is driven by the goal of promoting Jadescript towards a professional ecosystem by considering the preferences of professional programmers together with the well-known practical advantages of modern IDEs. The target IDE for the current implementation of the discussed productivity tools is the Eclipse IDE, which is a well-known and appreciated tool that served Java and JADE programmers for several years. However, the porting of the tools to other popular IDEs has already been considered.

Xtext was used as the basic building block of the internals of the Jadescript compiler. Therefore, this paper provides a brief overview of Xtext to detail its functionality and to explain the relevance of such a tool for Jadescript. Actually, Xtext is not only used to generate Java code from Jadescript code, but it is also used for the validation of Jadescript code and for the generation of errors, warnings, and recommendations for the programmer.

After the brief description of Xtext, the paper discusses the developed Eclipse IDE Plugin for Jadescript starting with a short digression on the token source system and its Xtext extension. Such an extension is of primary importance for the tools presented in this paper because it provides the basic support for semantically relevant indentation. Note that semantically relevant indentation plays an important role to greatly improve readability of Jadescript codes and to give a modern appeal to the language. The presented Eclipse IDE Plugin allows programmers to take advantage of all the features that the Eclipse IDE provides for other languages like Java. In particular, the Jadescript perspective provides a customised view of the Eclipse IDE specifically designed to accommodate the needs of Jadescript programmers. The Jadescript perspective customises the package explorer and the outline view of the current Jadescript code, and it provides the Jadescript code editor. The syntax highlighting support integrated with the Jadescript code editor further enhances the readability of Jadescript codes. Moreover, all the actions that the user can perform on mentioned user interface elements are tailored on the needs of a Jadescript programmer. Finally, new Eclipse wizards are provided to ease the creation of Jadescript projects.

The productivity tools discussed in this paper are intended to fulfil the need for a professional tool to match the expectations of ordinary Java programmers, and of JADE programmers, in particular. Nonetheless, such tools cannot be considered complete and further developments have been already planned. For example, Jadescript semantic classes, defined as part of the Xtext-based compiler, could be extended to implement advanced language tools like an improved validation system with quick fixes and auto-completion actions.

References

- [1] J. P. Müller, K. Fischer, Application impact of multi-agent systems and technologies: A survey, in: *Agent-Oriented Software Engineering*, Springer, 2014, pp. 27–53.
- [2] J. B. Larsen, Agent programming languages and logics in agent-based simulation, in: *Modern Approaches for Intelligent Information and Database Systems*, Springer, 2018, pp. 517–526.
- [3] B. Lutati, I. Gontmakher, M. Lando, A. Netzer, A. Meisels, A. Grubshtein, AgentZero: A framework for simulating and evaluating multi-agent algorithms, in: *Agent-Oriented Software Engineering*, Springer, 2014, pp. 309–327.
- [4] S. Monica, F. Bergenti, Location-aware JADE agents in indoor scenarios, in: *Proceedings of the 16th Workshop “From Objects to Agents”*, volume 1382 of *CEUR Workshop Proceedings*, RWTH Aachen, 2015, pp. 103–108.
- [5] S. Monica, F. Bergenti, Experimental evaluation of agent-based localization of smart appliances, in: *EUMAS 2016, AT 2016: Multi-Agent Systems and Agreement Technologies*, volume 10207 of *LNCS*, Springer, 2017, pp. 293–304.

- [6] S. Monica, F. Bergenti, Indoor localization of JADE agents without a dedicated infrastructure, in: *MATES 2017: Multiagent System Technologies*, volume 10413 of *LNCIS*, Springer, 2017, pp. 256–271.
- [7] S. Monica, F. Bergenti, A comparison of accurate indoor localization of static targets via WiFi and UWB ranging, in: *Trends in Practical Applications of Scalable Multi-Agent Systems*, the PAAMS Collection, 2016, pp. 111–123.
- [8] S. Monica, F. Bergenti, An optimization-based algorithm for indoor localization of JADE agents, in: *Proceedings of the 18th Workshop “From Objects to Agents”*, volume 1867 of *CEUR Workshop Proceedings*, RWTH Aachen, 2017, pp. 65–70.
- [9] S. Monica, F. Bergenti, An experimental evaluation of agent-based indoor localization, in: *Proceedings of the Computing Conference 2017*, IEEE, 2018, pp. 638–646.
- [10] S. Monica, F. Bergenti, Optimization based robust localization of JADE agents in indoor environments, in: *Proceedings of the 3rd Italian Workshop on Artificial Intelligence for Ambient Assisted Living (AI*AAL.IT 2017)*, volume 2061 of *CEUR Workshop Proceedings*, RWTH Aachen, 2017, pp. 58–73.
- [11] F. Bergenti, G. Caire, D. Gotta, An overview of the AMUSE social gaming platform, in: *Proceedings of the Workshop “From Objects to Agents” (WOA 2013)*, volume 1099 of *CEUR Workshop Proceedings*, RWTH Aachen, 2013.
- [12] F. Bergenti, G. Caire, D. Gotta, Agent-based social gaming with AMUSE, in: *Proceedings of the 5th International Conference on Ambient Systems, Networks and Technologies (ANT 2014) and 4th International Conference on Sustainable Energy Information Technology (SEIT 2014)*, *Procedia Computer Science*, Elsevier, 2014, pp. 914–919.
- [13] F. Bergenti, S. Monica, Location-aware social gaming with AMUSE, in: Y. Demazeau, T. Ito, J. Bajo, M. J. Escalona (Eds.), *Advances in Practical Applications of Scalable Multi-agent Systems. The PAAMS Collection: 14th International Conference, PAAMS 2016*, Springer International Publishing, 2016, pp. 36–47.
- [14] F. Bergenti, G. Caire, D. Gotta, Large-scale network and service management with WANTS, in: *Industrial Agents: Emerging Applications of Software Agents in Industry*, Elsevier, 2015, pp. 231–246.
- [15] F. Bergenti, M.-P. Gleizes, F. Zambonelli (Eds.), *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*, Springer, 2004.
- [16] K. Kravari, N. Bassiliades, A survey of agent platforms, *Journal of Artificial Societies and Social Simulation* 18 (2015) 11.
- [17] Y. Shoham, Agent-oriented programming, *Artificial Intelligence* 60 (1993) 51–92.
- [18] Y. Shoham, An overview of agent-oriented programming, in: J. Bradshaw (Ed.), *Software Agents*, volume 4, MIT Press, 1997, pp. 271–290.
- [19] M. Dastani, A survey of multi-agent programming languages and frameworks, in: *Agent-Oriented Software Engineering*, Springer, 2014, pp. 213–233.
- [20] F. Bergenti, G. Petrosino, Overview of a scripting language for JADE-based multi-agent systems, in: *Proceedings of the 19th Workshop “From Objects to Agents” (WOA 2018)*, volume 2215 of *CEUR Workshop Proceedings*, RWTH Aachen, 2018, pp. 57–62.
- [21] G. Petrosino, F. Bergenti, An introduction to the major features of a scripting language for JADE agents, in: *Proceedings of the 17th Conference of the Italian Association for Artificial Intelligence (AI*IA 2018)*, volume 11298 of *LNAI*, Springer, 2018, pp. 3–14.

- [22] F. Bergenti, S. Monica, G. Petrosino, A scripting language for practical agent-oriented programming, in: Proceedings of the 8th ACM SIGPLAN International Workshop on Programming Based on Actors, Agents, and Decentralized Control (AGERE 2018) at ACM SIGPLAN Conference Systems, Programming, Languages and Applications: Software for Humanity (SPLASH 2018), ACM Press, 2018, pp. 62–71.
- [23] F. Bellifemine, F. Bergenti, G. Caire, A. Poggi, JADE – A Java Agent DEvelopment Framework, in: Multi-Agent Programming, volume 25 of MASA, Springer-Verlag, 2005, pp. 125–147.
- [24] F. Bergenti, G. Caire, S. Monica, A. Poggi, The first twenty years of agent-based software development with JADE, *Autonomous Agents and Multi-Agent Systems* 34 (2020).
- [25] F. Bergenti, An introduction to the JADEL programming language, in: Proceedings of the 26th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), IEEE Press, 2014, pp. 974–978.
- [26] F. Bergenti, E. Iotti, S. Monica, A. Poggi, A case study of the JADEL programming language, in: Proceedings of the 17th Workshop “From Objects to Agents” (WOA 2016), volume 1664 of *CEUR Workshop Proceedings*, 2016, pp. 85–90.
- [27] F. Bergenti, E. Iotti, A. Poggi, Core features of an agent-oriented domain-specific language for JADE agents, in: Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection, Springer International Publishing, 2016, pp. 213–224.
- [28] F. Bergenti, E. Iotti, S. Monica, A. Poggi, Interaction protocols in the JADEL programming language, in: Proceedings of the 6th ACM SIGPLAN International Workshop on Programming Based on Actors, Agents, and Decentralized Control (AGERE 2016) at ACM SIGPLAN Conference Systems, Programming, Languages and Applications: Software for Humanity (SPLASH 2016), ACM Press, 2016, pp. 11–20.
- [29] F. Bergenti, E. Iotti, S. Monica, A. Poggi, Overview of a formal semantics for the JADEL programming language, in: Proceedings of the 18th Workshop “From Objects to Agents”, volume 1867 of *CEUR Workshop Proceedings*, RWTH Aachen, 2017, pp. 55–60.
- [30] F. Bergenti, E. Iotti, S. Monica, A. Poggi, A comparison between asynchronous backtracking pseudocode and its JADEL implementation, in: Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017), volume 2, SciTePress, 2017, pp. 250–258.
- [31] F. Bergenti, E. Iotti, S. Monica, A. Poggi, Agent-oriented model-driven development for JADE with the JADEL programming language, *Computer Languages, Systems & Structures* 50 (2017) 142–158.
- [32] R. H. Bordini, J. F. Hübner, M. Wooldridge, Programming multi-agent systems in AgentSpeak using Jason, volume 8, John Wiley & Sons, 2007.
- [33] A. S. Rao, AgentSpeak(L): BDI agents speak out in a logical computable language, in: European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Springer, 1996, pp. 42–55.
- [34] K. V. Hindriks, F. S. De Boer, W. Van der Hoek, J.-J. C. Meyer, Agent programming in 3APL, *Autonomous Agents and Multi-Agent Systems* 2 (1999) 357–401.
- [35] K. V. Hindriks, F. S. De Boer, W. Van Der Hoek, J.-J. C. Meyer, Agent programming with declarative goals, in: International Workshop on Agent Theories, Architectures, and Languages, Springer, 2000, pp. 228–243.

- [36] S. Rodriguez, N. Gaud, S. Galland, SARL: A general-purpose agent-oriented programming language, in: 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), volume 3, IEEE, 2014, pp. 103–110.
- [37] R. C. Cardoso, A. Ferrando, A review of agent-based programming for multi-agent systems, *Computers* 10 (2021) 16.
- [38] Eclipse IDE, <https://www.eclipse.org/ide>, 2021. Accessed on July 8th, 2021.
- [39] M. Eysholdt, H. Behrens, Xtext - Implement your language faster than the quick and dirty way tutorial summary, in: Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion, SPLASH '10, 2010.
- [40] IntelliJ IDEA, <https://www.jetbrains.com/idea>, 2021. Accessed on July 8th, 2021.
- [41] T. J. Parr, R. W. Quong, ANTLR: A predicated-LL(k) parser generator, *Software: Practice and Experience* 25 (1995).
- [42] T. J. Parr, K. Fisher, LL(*): The foundation of the ANTLR parser generator, in: Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2011.

On the Design of PSyKE: A Platform for Symbolic Knowledge Extraction

Federico Sabbatini¹, Giovanni Ciatto¹, Roberta Calegari² and Andrea Omicini¹

¹*Dipartimento di Informatica – Scienza e Ingegneria (DISI), ALMA MATER STUDIORUM—Università di Bologna, Italy*

²*Alma Mater Research Institute for Human-Centered Artificial Intelligence, ALMA MATER STUDIORUM—Università di Bologna, Italy*

Abstract

A common practice in modern explainable AI is to *post-hoc* explain black-box machine learning (ML) predictors – such as neural networks – by extracting *symbolic* knowledge out of them, in the form of either rule lists or decision trees. By acting as a surrogate model, the extracted knowledge aims at revealing the inner working of the black box, thus enabling its inspection, representation, and explanation.

Various knowledge-extraction algorithms have been presented in the literature so far. Unfortunately, running implementations of most of them are currently either proof of concepts or unavailable. In any case, a unified, coherent software framework supporting them all – as well as their interchange, comparison, and exploitation in arbitrary ML workflows – is currently missing.

Accordingly, in this paper we present PSyKE, a platform providing general-purpose support to symbolic knowledge extraction from different sorts of black-box predictors via many extraction algorithms. Notably, PSyKE targets symbolic knowledge in *logic form*, allowing the extraction of first-order logic clauses. The extracted knowledge is thus both machine- and human-interpretable, and can be used as a starting point for further symbolic processing—e.g. automated reasoning.

Keywords

explainable AI, knowledge extraction, interpretable prediction, PSyKE

1. Introduction

Artificial neural networks (ANN), support vector machines (SVM), and other data-driven predictors are nowadays among the most-used tools to face a wide range of different tasks involving machines learning (ML) from data [1]. In all those cases, the learning activity consists of tuning the parameters of predefined algorithms in order to maximise their predictive capability w.r.t. the data at hand.

The major drawback of state-of-the-art ML algorithms is that they are inherently *opaque*, meaning that they do not provide any intelligible representation of what they learn from data. This is why most of those algorithms are considered as black boxes (BB) which only represent

WOA 2021: Workshop “From Objects to Agents”, September 1–3, 2021, Bologna, Italy

✉ f.sabbatini@unibo.it (F. Sabbatini); giovanni.ciatto@unibo.it (G. Ciatto); roberta.calegari@unibo.it (R. Calegari); andrea.omicini@unibo.it (A. Omicini)

🌐 <http://federicosabbatini.apice.unibo.it> (F. Sabbatini); <https://about.me/gciatto> (G. Ciatto);

<http://robertacalegari.apice.unibo.it> (R. Calegari); <http://andreaomicini.apice.unibo.it> (A. Omicini)

🆔 0000-0002-0532-6777 (F. Sabbatini); 0000-0002-1841-8996 (G. Ciatto); 0000-0003-3794-2942 (R. Calegari); 0000-0002-6655-3869 (A. Omicini)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

knowledge in a *sub-symbolic* way. Nevertheless, despite their sub-symbolic operation may prevent human users from understanding *how* they work, BB – and, in particular, ANN – are being increasingly applied to support forecasting and decision making in many different fields – including, but not limited to, marketing, customer/user profiling, social networks, predictive maintenance, etc. – because of their unprecedented predictive performance.

There exist, however, critical applications where black-box predictions or recommendations are unacceptable: for instance, healthcare, finance and law domains, or any other area of knowledge where decision making may affect critical aspects of human lives—e.g. health, wealth, freedom, etc. In all those cases, it is of paramount importance to rely on *explainable* predictions, recommendations, or suggestions, in order to let humans retain accountability and liability over the decision or choices they make.

Many strategies can be exploited to pursue the purpose of explainability [2]. Some authors suggest for instance to *only* rely on *interpretable* algorithms [3] – such as generalised linear models, decision trees, etc. – to obtain data-driven solutions that are explainable by construction. However, this may hinder predictive performance in the general case, as it essentially cuts off most effective algorithms—e.g., ANN. Another strategy consists of deriving *post-hoc* explanations [4], aimed at reverse-engineering the inner operation of a BB so as to make it explicit. In this way, data scientists can keep using prediction-effective algorithms such as ANN, while still attaining high predictive performance. The focus of this paper is on the latter strategy.

Symbolic knowledge extraction (SKE) is among the most promising means to derive *post-hoc* explanations for sub-symbolic predictors. Roughly speaking, the main idea behind SKE is to enable the construction of a *symbolic* surrogate model mimicking the behaviour of a given predictor. There, symbols may consist of intelligible knowledge, such as *rule* lists or trees. Such rules can then be exploited to either derive predictions or to better understand the behaviour of the original predictor.

SKE has been applied, for instance, to credit-risk evaluation [5, 6, 7], healthcare – i.e., to make early breast cancer prognosis predictions [8] and to help the diagnosis and discrimination among hepatobiliary disorders [9] or other diseases and dysfunctions [10] –, credit card screening [11], intrusion detection systems [12], and keyword extraction [13].

Despite the wide adoption of SKE, however, a unified and general-purpose software technology supporting it is currently lacking. In other words, the burden of implementing SKE algorithms is currently on data scientists alone, who are likely to realise custom solutions on a per-need basis. Other than producing inertia w.r.t. the adoption of SKE in modern data, such a lack of viable technologies is somewhat anachronistic in the data-driven AI era, where a plethora of libraries and frameworks are flourishing, targeting all major programming paradigms and platforms, and making state-of-the-art machine learning algorithms easily accessible to the general public—cf. SciKit-Learn¹ for Python, or Smile² for the Java Virtual Machine (JVM).

Accordingly, in this paper we present the design of PSyKE, a general-purpose Platform for Symbolic Knowledge Extraction aimed at filling the gap between the current state of the art of SKE and the available technology. More precisely, PSyKE is conceived as an open library where different sorts of knowledge extraction algorithms can be realised, exploited, or compared.

¹<https://scikit-learn.org/stable> [Online; last accessed September 29, 2021].

²<https://github.com/haifengl/smile> [Online; last accessed September 29, 2021].

PSyKE supports rule extraction from both classifiers and regressors, and makes the extraction procedure as transparent as possible w.r.t. the underlying BB, depending on the particular extraction procedure at hand. Notably, it also supports the extraction of first-order logic (FOL) clauses, with the twofold advantage of providing human- and machine-interpretable rules as output. These can then be used as either an explanation for the original BB, or as a starting point for further symbolic computations. More precisely, the current implementation of PSyKE outputs logic programs [14, 15], expressed in Prolog syntax [16].

Furthermore, to demonstrate the versatility of PSyKE, we present a number of experiments involving rule extraction on a classification task performed on the Iris data set. In particular, we exemplify our framework against various BB predictors, and carry out a comparison between different extraction procedures applied to the same task. The comparison takes into account the fidelity of the extracted rules (w.r.t. the original BB) and their accuracy w.r.t. the data.

Accordingly, the remainder of this paper is organised as follows. Section 2 describes the state of the art for SKE as well as some background notion to fully understand the work. Section 3 presents the design of PSyKE, while in Section 4 some use cases showing how PSyKE can be exploited are reported. Conclusions are drawn in Section 5.

2. State of the Art

In this section we firstly overview the state of the art for symbolic knowledge extraction (Section 2.1). Then, we delve into the details of a selection of extraction algorithms—namely, the ones PSyKE implementation currently supports (Section 2.1.1–Section 2.1.3). The algorithm selection is performed by keeping *variety* (rather than exhaustivity) in mind, so as to demonstrate the operation and versatility of PSyKE. In particular, our aim is to exemplify the many application scenarios that a data scientist may meet—e.g., extraction from either classifiers or regressors, trained on either categorical or continuous data.

Finally, we briefly outline the currently-available software object-oriented frameworks for ML (Section 2.2). The overview is meant to make the paper self-contained, given that one of these frameworks provides PSyKE with pure ML functionalities—in particular, the design of PSyKE assumes basic classification or regression support to be available as a software library.

2.1. Knowledge Extraction

According to [17], a computational system is considered *interpretable* if human beings can easily understand its operation and outcomes. The majority of modern ML predictors, however, sacrifice interpretability to enhance the predictive performance, thus becoming increasingly complex. They do so by merely focusing on learning highly-predictive – yet *sub-symbolic* – input-output relations from data, while neglecting any attempt to make such relations *symbolic*, i.e., intelligible for human. For this reason, ML algorithms are often called *black boxes* [18].

To mitigate interpretability issues without sacrificing predictive performance, a number of authors from the XAI community have proposed means to produce *ex-post* explanations for sub-symbolic predictors—most notably, ANN and SVM. Explanations, in this case, consist of *surrogate* predictors trained to mimic the ones to be explained, as closely as possible.

In practice, among the manifold proposals, some authors describe methods to extract if-then-else rules [19, 20, 21], whereas others propose methods extracting decision trees [22]. While the shape of the extracted knowledge may vary from an extraction procedure to another, all the proposed methods share the trait of extracting *symbolic* (i.e. human-intelligible) knowledge out of *sub-symbolic* ML predictors. Given a trained predictor and a knowledge-extraction procedure applicable to it, the extracted rules/trees act as *explanations* for that predictor – or as a basis to build some –, provided that they retain high *fidelity* w.r.t. the underlying predictor [17]. The extracted knowledge may then enable further manipulations, such as merging the *know-how* of two or more BB models [23].

According to [24], knowledge extraction methods can be categorised along three orthogonal dimensions, namely: (i) the sort of learning tasks they support, (ii) the shape of the symbolic knowledge they produce, (iii) their *translucency*—i.e., the sort of BB algorithms they can extract symbols from.

About item (i), one can distinguish among algorithms targeting classification tasks, regression tasks, or both. In other words, some extraction algorithms can only deal with BB classifiers – e.g. Rule-extraction-as-learning [19] (REAL, henceforth), TREPAN [22] and others [25, 26] –, while others can only deal with BB regressors – such as ITER [20], GridEx [21], REFANN [27], ANN-DT [28] and RN2 [29] –, and only a few can handle both—such as G-REX [30] and CART [31]. Notably, virtually all extraction methods proposed so far are tailored on *supervised* machine learning. To the best of our knowledge, no rule extraction procedure has been proposed targetting unsupervised or reinforcement learning tasks.

As far as item (ii) is concerned, decision rules [32, 33, 34] and trees [35, 36] are the most widespread human-understandable shapes for extracted knowledge, thus most methods produce one of these two structures. In both cases, decision rules or nodes are expressed in terms of the same input/output data types the original BB has been trained upon. So, for instance, an extraction procedure processing a BB classifier for N -dimensional numerical data, over K classes, will likely output rules/trees involving one or more *predicates* over N input variables x_1, \dots, x_n and K possible outcomes. In any case, however, extraction algorithms are further categorised w.r.t. the particular sort of predicates their output rules/trees may contain. Accordingly, conjunctions/disjunctions of inequality (e.g. $x_i \geq c$), or interval inclusion/exclusion expressions (e.g. $x_i \in [l, u]$) are commonly exploited for numerical data, while equality (e.g. $x_i = c$) or set-inclusion $x_i \in \{c_1, c_2, \dots\}$ expressions may be exploited for categorical data. Finally, M -of- N rules are yet another possible choice in the case of boolean data.

The translucency dimension [37] from item (iii) refers to the need/capability of the extraction procedure to “look into” the *internal* structure of the underlying BB—i.e., to what extent it has to be taken into account during the extraction procedure. There are two major ways for categorising knowledge extractors w.r.t. translucency. During the extraction process, *decompositional* extractors may take into account the internal structure of the BB they operate upon, while *pedagogical* ones do not. For this reason, pedagogical approaches are usually more general – despite potentially less precise –, thus they can be applied to every BB predictor regardless of its kind, structure, and complexity.

The quality of knowledge-extraction procedures is evaluated through different indicators depending on the task to solve, for instance, fidelity and predictive performance measurements [38]. In particular, the former indicates how well the extracted knowledge mimics the underlying

Table 1

Summary of the knowledge-extraction algorithms supported by PSyKE.

Extraction Algorithm	Task	Translucency	Required Features	Knowledge Shape	Exhaustive
REAL	Classification	Pedagogical	One-hot encoded	Rule list	-
TREPAN	Classification	Pedagogical	One-hot encoded	Decision tree	✓
ITER	Regression	Pedagogical	Continuous	Rule list	-
GridEx	Regression	Pedagogical	Continuous	Rule list	✓
CART	Classification and regression	Pedagogical	Continuous or one-hot encoded	Decision tree	✓

black-box predictions, whereas the latter measures the explainer predictive power w.r.t. the data. In all cases, measurements should be taken via the same scoring function used for assessing the BB performance—which in turn depends on the task it performs. In the particular case of black-box classifiers, examples of performance measurements are accuracy, precision, recall, and F1-score; for BB regressors, the mean absolute/squared error (MAE/MSE) and the R^2 scores could be exploited.

In the following, we provide a more detailed description of some extraction procedures currently supported in the PSyKE framework, grouped by the task they address—i.e. classification, regression, or both of them. All the algorithms are *pedagogical*, thus they only rely on the BB inputs and outputs, and do not inspect the inner structure of the underlying BB. This is why they can be applied to any kind of arbitrarily-complex BB. In any case, structured data are required. The same algorithms are summarised in Table 1.

2.1.1. Extraction from Classifiers

Rule-extraction-as-learning REAL [19] is a pedagogical algorithm to extract conjunctive rules from trained BB classifiers by using a learning process driven by sampling and queries. Output rules can be either *if-then* or *M-of-N* rules. An example of *if-then* output rule is the following: *Output class is C if {X, Y, Z} are True and {U, V} are False*, where U, V, X, Y, Z are one-hot encoded input features. Such features are *True* if they are 1, *False* otherwise.

Output rules are disjunctive normal form expressions; each term is the conjunction of a data set feature subset, adequately generalised by dropping each non-discriminant antecedent. REAL cannot handle real-valued features, but only binary ones.

TREPAN TREPAN [22] is a pedagogical algorithm able to extract symbolic and comprehensible model representations from trained classifier BB by inducing a decision tree approximating the BB represented concept. It usually maintains high fidelity levels w.r.t. the underlying BB while being comprehensible and accurate. It is general in its applicability and well scalable with complex models or problems. However, as REAL, it cannot be applied to real-valued features.

In Figure 1 an example of output tree is reported. Internal nodes are represented by squares, while leaves are circles. Variables reported inside the internal nodes are the split criteria for the subtree creation. C_i are the class labels corresponding to each leaf.

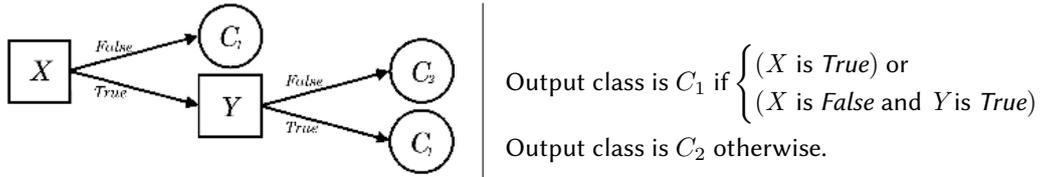


Figure 1: Example of TREPAN output tree (left) and corresponding rules (right).

2.1.2. Extraction from Regressors

ITER ITER [20] is a pedagogical algorithm for building predictive rules from trained BB regressors of any kind. Its main idea is to iteratively expand a number of hypercubes until they cover the whole input space. Each of them is finally converted into an *if-then* rule of the following format: *Output constant is C if $X_1 \in [l_1, u_1]$ and ... and $X_k \in [l_k, u_k]$* , where l_i and u_i are the lower-bound and the upper-bound for variable X_i . There, the preconditions of the rule describe a k -dimensional hypercube. Indeed, ITER supports continuous input features, differently from the other algorithms presented so far.

GridEx GridEx [21] is another pedagogical extraction algorithm for regressors; it is an extension of ITER aimed at overcoming its major drawback: non-exhaustivity. GridEx adopts a top-down approach to iteratively partition the input feature space in a user-defined number of hypercubes or in an automatic way accordingly to a user-defined strategy based on feature importance. As ITER, GridEx produces *if-then* rules and only accepts data sets with real-valued input features, but it is always exhaustive by design. Since the procedure associates each hypercube to a rule, a merging phase is performed after every iteration as an optimisation to reduce the number of rules.

2.1.3. General-Purpose Extractors

CART CART [31] is an algorithm for building decision trees that can be used to face both classification and regression tasks. CART is not properly a knowledge-extraction procedure, but from its output tree it is straightforward to obtain a rule tree, since each node of the CART tree corresponds to a constraint on a certain feature and thus each path from the root to a leaf is a single complete classification or regression rule. The algorithm can be summarised via the following instructions: (i) initialise the tree root node; (ii) find optimal splits and add new internal nodes and leaves accordingly; (iii) stop the algorithm based on one or more criteria—e.g., leaf number or tree depth. Pruning algorithms can be applied to reduce the number of leaves.

2.2. Object-Oriented Programming Frameworks for ML

In order to make ML solutions easily available, a number of frameworks – especially exploiting object-oriented programming (OOP) – have been developed. Such frameworks usually provide users with powerful abstractions for modelling BB models as well as for performing data set pre-processing, feature engineering and predictive performance measurements. Among the most supported ML models, there are ANN, SVM, and decision trees for both classification and

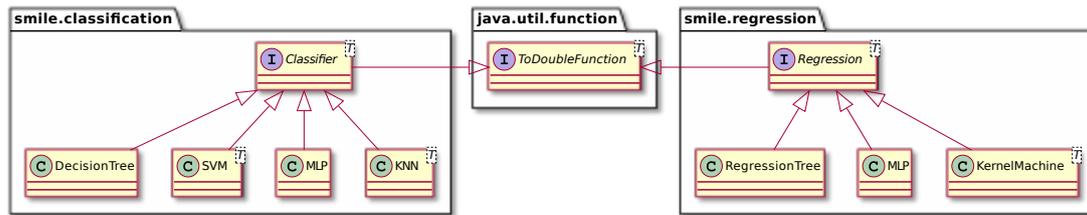


Figure 2: Overview on the API of the Smile library supporting classification and regression tasks.

regression tasks. The most complete frameworks also provide utilities packages to ease the data set reading from (and writing into) files and to perform feature selection, beyond many other tools for natural language processing, linear algebra, and data visualisation. Examples of state-of-the-art OOP frameworks for ML are SciKit-Learn [39] for Python and Smile for the JVM. We adopted the latter for the design and development of PSyKE because of its JVM support.

Smile (Statistical Machine Intelligence and Learning Engine) is defined as “a fast and comprehensive machine learning engine”³ for Java, Scala, and Kotlin. It is worthwhile to notice that the package provides data types for easily managing data set, feature vectors, and tuples (intended as data set columns and rows, respectively) other than all the aforementioned tools.

Figure 2 depicts a (partial) UML class diagram representing the major interfaces and classes composing Smile’s supervised learning API. Package names are explicitly indicated to avoid confusion between homonymous classes. Notably, each kind of ML predictor has a dedicated class and each class implements either the `Classifier` or the `Regression` interface. Both interfaces descend from the `ToDoubleFunction` interface, which is, therefore, the most adequate type to represent any supervised ML predictor. This design lets developers easily build more complex concepts over the packages offered by Smile—as in the PSyKE prototype. Of course, the same design could be replicated on different platforms (e.g. Python) and libraries (e.g. SciKit-Learn)—as long as they provide similar API to train and use ML classifiers or regressors.

3. PSyKE

PSyKE is a software library providing general-purpose support to the extraction of logic rules out of BB predictors by letting users choose the most adequate extraction method for the task and data at hand. PSyKE exposes a unified API covering virtually all extraction algorithms targeting supervised learning tasks. Currently, the implementation of PSyKE involves several interoperable, interchangeable, and comparable extraction procedures – namely, the ones mentioned in Section 2.1 –, granting access to state-of-the-art knowledge-extraction algorithms to both researchers and data scientists. PSyKE is conceived as an open-ended project, which can be exploited to design and implement new extraction procedures behind a unique API.

Essentially, PSyKE is designed around the notion of *extractor*, whose API is depicted in Figure 3. Within the scope of PSyKE, an extractor is any algorithm accepting a ML predictor – either a classifier or a regressor – as input, and producing a *theory* of logic rules as output.

³<https://haifengl.github.io> [Online; last accessed September 29, 2021].

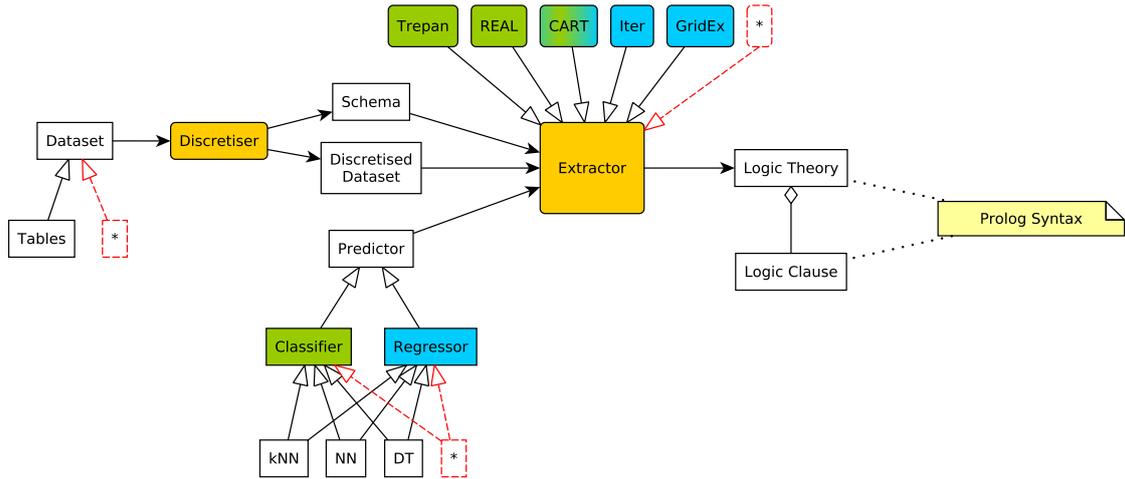


Figure 3: PSyKE design.

To perform their job, PSyKE extractors require additional information about the data set the input predictor has been trained upon. In the general case, such information consists of the data set itself and its schema—i.e., a formal description of the names and the data types of all features characterising the data set itself. More precisely, data sets are required to let extraction procedures inspect BB behaviour – and therefore build the corresponding output rules –, whereas schemas are required to let (i) the extraction procedure take informed decisions on the basis of the feature *types*, (ii) the extracted knowledge be clearer by referring to the feature *names*. For all these reasons, extractors expect a data set and its schema metadata to be provided in input as well.

Many extraction procedures can operate on discrete/binary data only. This is commonly made necessary by the shape of the extracted rules—which consists of simple predicative statements about some feature value. However, it is also very common in data science to meet data sets involving continuous attributes as well. Accordingly, extracting rules out of predictors trained on continuous data may be troublesome in the general case. To circumvent this issue, PSyKE also provides some facilities aimed at discretising (or binarising) data sets including continuous (or categorical) data. When these are in place, extractors should be provided with the discretised/binarised schema as well, to be able to produce the clearest rules possible.

Accordingly, in the rest of this section we detail (i) the general design of the PSyKE library and API, (ii) the one-hot encoding facilities, (iii) the general shape of the extracted logic theory.

3.1. General API

As depicted in Figure 4, a pivotal role in the design of PSyKE is played by the Extractor interface – reported in Figure 4 –, defining the general contract of any knowledge-extraction procedure. More precisely, it leverages on the notions of DataFrame and Theory, borrowed from Smile and 2P-KT [40, 41], respectively. All the PSyKE extractors expose (i) a method for extracting an explainable theory from a BB model and (ii) a method to make predictions by

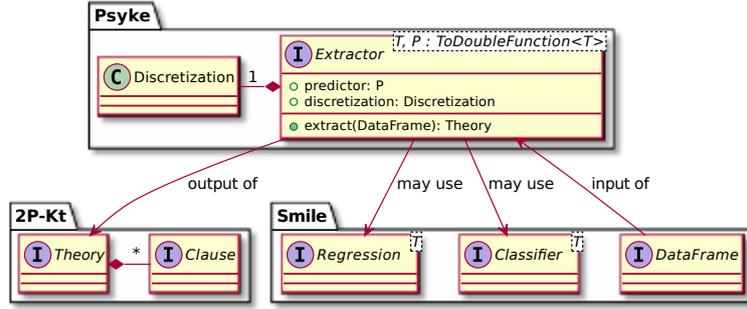


Figure 4: PSyKE's Extractor interface

using the extracted rules. The common API makes it possible to switch between different PSyKE extractors with no need of major changes in the code.

3.2. Discretisation

A large number of the knowledge-extraction procedures – in the same way as many ML algorithms – require either a discrete or binary input space—i.e. all input features must be either categorical or one-hot encoded, respectively. For instance, REAL and TREPAN require exclusively one-hot encoded data, whereas ITER and GridEx require continuous data. CART can accept both continuous and one-hot encoded features, but not categorical ones. Unfortunately, most real-world applications are described by real-valued variables and measurements, thus making the application of such algorithms impractical. The general way to overcome this limitation is to rely on some discretisation/binarisation method among the many available in the literature—e.g., [42, 43, 44, 45, 46, 47, 48].

Briefly speaking, discretisation is the process of transforming a datum from some continuous space $I \subseteq \mathbb{R}$ into a discrete space $\{I_1, \dots, I_n\}$ such that $\forall i, j = 1, \dots, n: I_i \subset I \wedge I \equiv \bigcup_i I_i \wedge i \neq j \Leftrightarrow I_i \cap I_j = \emptyset \wedge i < j \Leftrightarrow \forall x \in I_i, \forall y \in I_j : x < y$. Similarly, binarisation (a.k.a. one-hot encoding) is the process of transforming a datum from some discrete space $X = \{x_1, \dots, x_n\}$ into a binary space $B = \{b_1, \dots, b_n\}$ where for each $i = 1, \dots, n: b_i$ is 1 if the datum is equal to x_i , 0 otherwise. Of course, these methods imply a considerable increase in the dimensionality of a data set—e.g., one-hot encoding makes categorical attributes with 4 distinct values be converted into 4 different boolean features. This is far from being an issue: in some cases, it is possible to achieve even better classification performances by using discretised attributes rather than continuous, as demonstrated in [49].

PSyKE provides different procedures to manipulate input features: (i) a discretisation for continuous features, mapping real intervals into categorical features, and (ii) a one-hot encoding for categorical features, mapping exact values to boolean features. Notably, PSyKE traces the input feature transformations by creating a data structure that associates the initial name of the attribute and the newly created features with the corresponding constraints. An example of PSyKE binarisation and corresponding output data structure is reported in Figure 5. This example considers the *petal length* attribute of the Iris data set and adopts the default supervised discretisation method available in our framework. The initial continuous feature values are

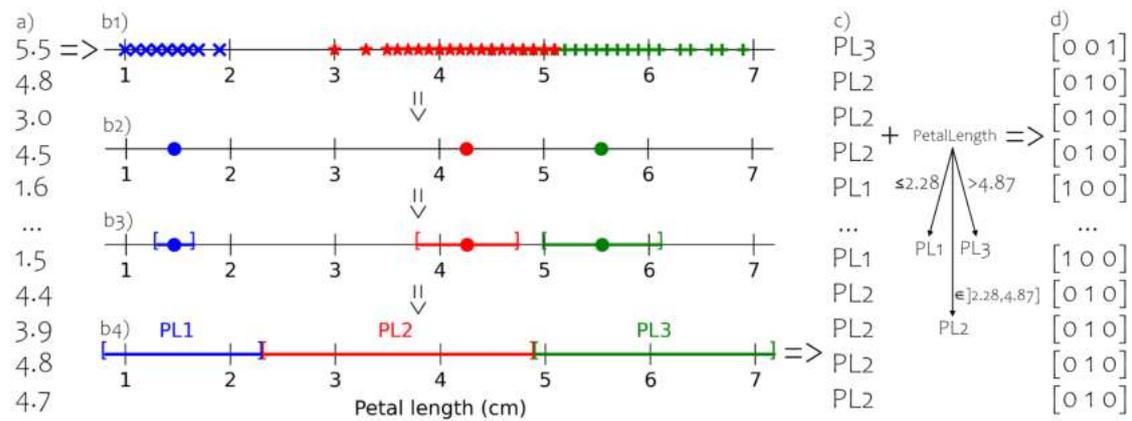


Figure 5: PSyKE discretisation and binarisation procedure.

labelled in Figure 5 with a), and graphically represented in the b1) plot. PSyKE discretisation algorithm consists in calculating the mean attribute value and the corresponding standard deviation *for each data set class*. An interval is then initialised for each class, with lower and upper bounds equal to the mean value—cf. plot b2). Each interval is iteratively expanded until convergence—i.e., when the whole feature space is covered without overlapping intervals. To achieve this, during every iteration all the intervals are symmetrically expanded of a value equal to the corresponding standard deviation – in each direction, as in plot b3) –, in order to create intervals with adaptive size. An expansion is inhibited when (i) the lower (upper) bound of an interval exceeds the minimum (maximum) value of the feature space, or (ii) two adjacent intervals are overlapping. In the first case, the feature minimum (maximum) value is taken as the final interval lower (upper) bound. In the second case, the overlapping intervals are only expanded up to the mean value between their respective boundaries. When all intervals have been calculated – cf. plot b4) – the continuous attribute values are converted accordingly into categorical values. The discretised output values are labelled in the example with c). In this step PSyKE also produces a data structure for keeping the discretisation details, to be able to produce more compact rules during the extraction procedures. The last step – labelled with d) – is the one-hot encoding of the discrete values into arrays of binary data—i.e., the unique format accepted by several extraction procedures, such as REAL and TREPAN.

3.3. Output rules

PSyKE extractors output knowledge in the form of logic theories – i.e. lists of Horn clauses –, notably in Prolog syntax. We choose the Prolog syntax to make them simultaneously interpretable by both humans and machines. More precisely, PSyKE output theories are structured as lists of Prolog rules or facts. Rule heads are $(n + 1)$ -ary predicates, where n is the number of input features in the dataset. These predicates carry n variables – i.e., one for each input feature – and either a constant or a list – i.e., the output value(s) – as argument. Predicate names recall the classification/regression under study. Without lack of generality, in the following we assume the case under study to involve *mono-dimensional* classification/regression tasks.

Rule bodies can be empty – if rules are *facts* – or conjunctions of literals where each literal is a predicate expressing inequality, equality, or interval inclusion between attribute actual values and fixed constants calculated through the extraction process. Accordingly, a rule-extraction procedure targeting a mono-dimensional classification or regression *task* on a data set having n input features and m relevant output values, shall output theories of the following form:

$$\begin{aligned} \langle task \rangle(x_1, \dots, x_n, y_1) & :- p_{1,1}(\bar{x}), \dots, p_{n,1}(\bar{x}). \\ \langle task \rangle(x_1, \dots, x_n, y_2) & :- p_{1,2}(\bar{x}), \dots, p_{n,2}(\bar{x}). \\ & \vdots \\ \langle task \rangle(x_1, \dots, x_n, y_m) & :- p_{1,m}(\bar{x}), \dots, p_{n,m}(\bar{x}). \end{aligned}$$

where (i) *task* is the $(n + 1)$ -ary relation representing the classification or regression task at hand, (ii) each x_i is a logic variable named after the i^{th} input attribute of the currently available data set, (iii) \bar{x} is the n -tuple x_1, \dots, x_n , and (iv) each $p_{i,j}$ is either a n -ary predicate expressing some constraint about one, two or more variables, or the `true` literal—which can be omitted. Notice that, in classification tasks, the total amount of rules (m) may still be greater than the total amount of classes (k), as there may be more than one rule for the same class.

Currently, the supported sorts of predicates in rules bodies – i.e. the admissible shapes for each $p_{i,j}$ – are as follows:

equality involving a single variable and a constant—e.g. $x = c$, where c is a constant of any sort (possibly, a number)⁴

inequalities involving a single variable and a constant—e.g. $x \geq c$

interval inclusion involving a single variable and two constants—e.g. $x \text{ in } [l, u]$, where $l, u \in \mathbb{R}$ and $l < u$

interval exclusion like the above, but negated—e.g. $x \text{ not_in } [l, u]$

M-of-N involving N variables—e.g. $\text{at_least}(M, [x_1, \dots, x_N])$, where $M, N \in \mathbb{N}_{\geq 0}$

Despite many other forms can be adopted for the output theories, we argue the proposed one is a good trade-off among human and machine interpretability. In fact, rules of this form are well-formed logic programs, which may be executed by a logic reasoner—such as a Prolog interpreter. Furthermore, the proposed form is open to many sorts of post-processing. For instance, recurrent conjunctions of predicates in rules bodies may be factorised into their own general-purpose rules. Similarly, redundant or cumbersome sub-expressions may be simplified.

However, the proposed form is far from perfection: we plan to explore alternative directions in the future. Noticeably, using Prolog syntax does not impose exploiting also its semantics—i.e., different interpreters can be exploited over such Prolog rules. For instance, on the one side, by exploiting a Prolog interpreter, rules are interpreted as functional (one-way)—meaning that it is possible to compute a prediction given an assignment of all input variables, but it is not possible to generate a correct assignment of those input variables given the expected prediction alone. On the other side, a Constraint Logic Programming solver [50, 51] may interpret the same rules as constraints, and compute coherent assignments for any subset of both input and output variables—providing rules with a relational (two-ways, generative) semantics.

⁴The same result could be attained by allowing constants in rules heads.

4. Case Study: The Iris Data Set

Here we exemplify the effectiveness and versatility of PSyKE by describing its exploitation in a toy scenario. In particular we exploit PSyKE to extract Prolog rules on a number of classifiers trained on the well-known Iris data set.⁵ Notably, the Iris data set contains 150 rows describing as many individuals of the Iris flower. For each exemplary, 4 continuous input features – petal and sepal width and length – are recorded, other than a categorical class label—i.e. which particular sort of Iris plant the exemplary has been classified as. There are three particular sub-sorts of Iris in this data set – namely, Setosa, Virginica, and Versicolor –, and the 150 examples are evenly distributed among them—i.e. there are 50 instances for each class.

The experimental setting is as follows. First, we train 3 different sorts of classifiers on the Iris data set—namely, a k-nearest-neighbors (kNN), a multi-layer perceptron (MLP), and a decision tree (DT). Then we let PSyKE extract logic rules out of these classifiers using as many extraction procedures. In particular, we rely on REAL, TREPAN, and CART. A portion (50%) of the original data set – namely, the test set – is put aside *before* training to later enable the evaluation of the extracted rule predictive performance.

Accordingly, within the scope of this experiment, we rely on *accuracy* as the preferred metric for both predictive performance and fidelity—where the former measures how good a classifier or the corresponding extracted rules are in classifying Iris instances in absolute terms, while the latter measures the adherence of the extractor output rules w.r.t. the original classifier.

4.1. The experiment

Let us assume the Iris data set can be loaded from a CSV file via a Kotlin script using Smile. The Iris data set only contains continuous features. Therefore, CART is the only algorithm that can be *directly* applied to it, whereas REAL and TREPAN can only operate on binary data. Accordingly, PSyKE provides a simple two-step procedure to binarise the data, involving both discretisation and one-hot encoding: a data set can be discretised, one-hot encoded, and split into training and test set via a couple of instructions, providing the percentage of samples to be taken apart from the whole data set to attain the test set. As the next step, we train 3 different classifiers on the training set—namely a kNN, a MLP, and a DT.⁶ Finally, in the following paragraphs, we show how rules can actually be extracted and what their ultimate shape actually is.

4.1.1. REAL

The PSyKE REAL algorithm can be applied to any Smile `Classifier` model parametrised with a `DoubleArray`—e.g., the aforementioned MLP and kNN are suitable, whereas the DT is not. The extracted theory is dependent from the training set; different training instances can produce different rules, resulting in slight variations also in the output theory complexity—intended as number of clauses and terms. An example is reported in the following:

⁵<https://archive.ics.uci.edu/ml/datasets/iris> [Online; last accessed September 29, 2021].

⁶Code of experiments is available at <https://github.com/sabbatinif/PSyKE>. Please consider reading the documentation of Smile for a more detailed explanation of the model-specific parameters: <https://haifengl.github.io/classification.html> and <https://haifengl.github.io/regression.html>.

```

1 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, setosa) :-
2   PetalWidth =< 0.65.
3 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, versicolor) :-
4   PetalWidth in [0.65, 1.64].
5 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, virginica) :-
6   PetalWidth > 1.64.

```

The theory produces an input space partitioning as reported in Figure 6c. It is worthwhile to notice that – especially with more complex data sets – the partitioning could be non-exhaustive—i.e., the logic rules could be unable to classify some samples.

4.1.2. TREPAN

PSyKE provides a TREPAN algorithm applicable under the same constraint described above for REAL and also its output rules can vary with different training sets. Differently from REAL, TREPAN accepts as input 3 optional parameters stating the minimum number of samples to consider for performing further splits (`minExamples`, default: 0), the maximum depth of the produced tree (`maxDepth`, default: 0, i.e. no constraints), and the criterion to adopt for the best split selection (`splitLogic`). A default logic is chosen if not otherwise specified. An example of output theory is reported in the following:

```

1 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, setosa) :-
2   PetalLength =< 2.28.
3 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, virginica) :-
4   PetalLength > 2.28, PetalWidth not_in [0.65, 1.64].
5 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, versicolor) :- true.

```

This theory produces an input space partitioning as reported in Figure 6d. In this case, the partitioning is always exhaustive.

4.1.3. CART

CART is the third algorithm included in PSyKE to tackle classification tasks. It is the only procedure applicable to a `DecisionTree` classifier and it does not require any extra parameter, since the extraction only relies on the tree structure of the `DecisionTree`—that is, all the parameters have to be tuned during the `DecisionTree` creation and training. The `DecisionTree` can accept both one-hot encoded and continuous features. In the same way as many other algorithms, CART is able to achieve comparable or even better results when relying on a good discretisation/one-hot encoding technique. In the following an example of theory obtained with continuous features is reported:

```

1 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, setosa) :-
2   PetalLength =< 2.75.
3 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, versicolor) :-
4   PetalLength > 2.75, PetalLength =< 4.85.
5 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, virginica) :-
6   PetalLength > 2.75, PetalLength > 4.85.

```

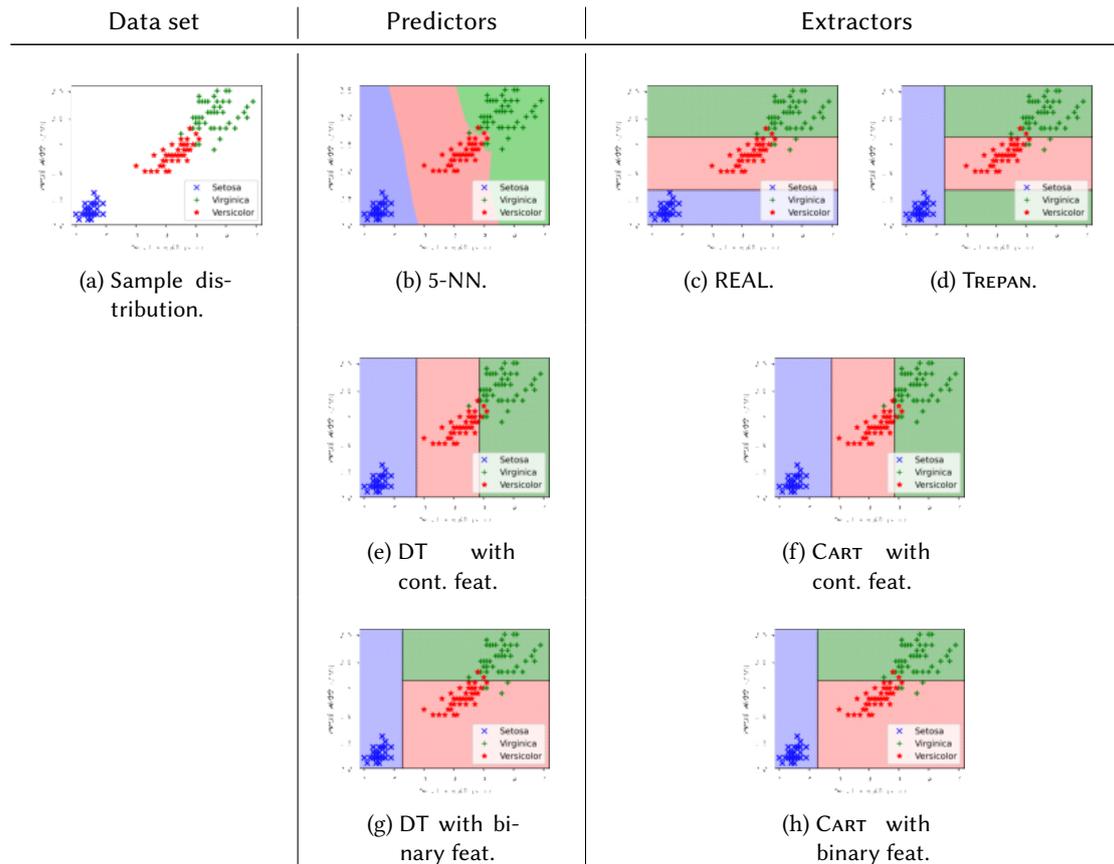


Figure 6: Comparison between Iris data set input space partitionings performed by the algorithms implemented in PSyKE. Only the two most relevant features are reported—i.e., petal width and length.

Figure 6f reports the corresponding input space partitioning. The output rules are always exhaustive. The same data set, but previously one-hot encoded, leads to the following theory and to the partitioning reported in Figure 6h:

```

1 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, versicolor) :-
2   PetalLength > 2.28, PetalWidth =< 1.64.
3 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, virginica) :-
4   PetalLength > 2.28, PetalWidth > 1.64.
5 iris(SepalLength, SepalWidth, PetalLength, PetalWidth, setosa) :-
6   PetalLength =< 2.28.

```

4.2. Results

In the following we report the results of REAL, TREPAN, and CART applied to the Iris data set. All the results are resumed in Figure 6. Figure 6a reports the Iris data set sample distribution in the input space, with emphasis only on the two most relevant features—i.e., petal width and length. Column *Predictors* represents the ML step of the process. Accordingly, Figure 6b,

Table 2

Comparison between accuracy and fidelity measurements with different combinations of extractor algorithms and underlying models.

Predictor Type	Accuracy	Extractor		
		Algorithm	Fidelity	Accuracy
5-NN	0.94	REAL	0.98	0.95
		TREPAN	0.96	0.96
MLP	0.92	REAL	0.95	0.95
		TREPAN	0.99	0.92
DT (continuous features)	0.92	CART	1.00	0.92
DT (binarised features)	0.96	CART	1.00	0.96

Figure 6e and Figure 6g represents the decision boundaries of a 5-NN predictor and 2 decision trees trained with a half of the data set samples, respectively. The first DT is trained with the original Iris data set; the other with the binarised version. Finally, column *Extractors* represents the PSyKE output. In particular, different extraction procedures – namely, REAL and TREPAN – applied to the 5-NN are depicted in Figure 6c and Figure 6d, respectively – and CART extraction applied to decision trees is depicted in Figure 6f and Figure 6h.

It is worth noticing that (i) CART – for its design – always produces partitionings equivalent to those of the underlying DT, and (ii) all extractor output partitionings are different – namely, a procedure only uses the petal width attribute, another one uses only the petal length and the remaining use both input features –, but all solutions share a similar predictive performance.

Omitted in Figure 6, the results of REAL and TREPAN applied to a MLP are presented in Table 2, where a numerical assessment of the aforementioned predictors and extractors is reported. Values are averaged upon 25 executions, each one with different random train/test partitionings, but same test set percentage and same parameters for predictors and extractors. The table reports the underlying predictor accuracy as well as the fidelity and accuracy of the extraction procedure. We plan to enhance comparisons between different extractors through fidelity assessments carried out by measuring the decision boundary overlapping regions. From Table 2 it can be noticed that the CART extractor always has a fidelity of 1.0, since it only inspects the underlying decision tree nodes to build its output rules without any knowledge loss. This implies that PSyKE CART is an equivalent but explainable alternative to `Smile DecisionTree` model, always producing the same output predictions. As for the other extractors, both REAL and TREPAN are able to achieve good results in terms of fidelity and accuracy – always above 0.9 in our experiments – in some cases even with a better performance w.r.t. the original model.

5. Conclusions

In this paper we present the design of PSyKE, a new general-purpose platform supporting symbolic knowledge extraction from opaque ML predictors. PSyKE offers many comparable and interchangeable extraction procedures providing as output first-order logic clauses. It can be

exploited in the majority of supervised learning tasks—i.e., classification and regression tasks.

In the future we plan to enrich PSyKE with other state-of-the-art extraction algorithms, comparison metrics between the implemented procedures, and other utilities—i.e., discretisation strategies. We also plan to explore other formalisms to present output rules – e.g. the ProbLog syntax to introduce the concept of probabilistic rule –, as well as other representations and extraction procedures which are better suited to manage data sets involving a wide number of features.

From a research perspective, we aim at further investigating the effectiveness of PSyKE in running EU projects, like StairwAI⁷ and EXPECTATION [52].

StairwAI is an H2020 project aimed at providing a service layer for the AI-on-demand platform,⁸ with the purpose of aiding both individual and companies to (i) find the most adequate AI asset for their needs – requiring mapping of use cases to proper AI assets –, and (ii) experimenting selected AI assets on custom data and on specific problems, using the platform itself—thus requiring tools for predicting the hardware resources needed for running the corresponding software.

EXPECTATION is a CHIST-ERA IV project⁹ aimed at exploring the provisioning of *personalised* explanations for ML techniques by combining SKE and multi-agent-based negotiation and argumentation. There, symbolic knowledge is expected to act as the *lingua franca* among many heterogeneous ML-based predictors – possibly trained on different data sets, via different algorithms, at different locations –, hosted by as many software agents. Personalisation and predictive accuracy are therefore attained by combining the symbolic knowledge extracted by several agents. The combination takes advantage of negotiation and argumentation techniques, possibly involving the users themselves.

Both projects massively rely on sub-symbolic AI, and in both cases the need of making sub-symbolic knowledge explainable is prominent. PSyKE could then be applied to extract logic rules and reveal information about the path that leads to a certain prediction—in the explanation perspective. Since PSyKE currently works as a distiller of knowledge, further investigation will be devoted to the explanation of a single outcome (prediction of a model). Moreover, it could be interesting to compare results with those obtained by directly learning a symbolic model .

Acknowledgments

This paper has been partially supported by (i) the European Union’s Horizon 2020 research and innovation programme under G.A. no. 101017142 (StairwAI project), and by (ii) the CHIST-ERA IV project CHIST-ERA-19-XAI-005, co-funded by the EU and the Italian MUR (Ministry for University and Research).

⁷<https://cordis.europa.eu/project/id/101017142> [Online; last accessed September 29, 2021].

⁸<https://cordis.europa.eu/project/id/825619> [Online; last accessed September 29, 2021].

⁹<https://www.chistera.eu/projects/expectation> [Online; last accessed September 29, 2021].

References

- [1] A. Rocha, J. P. Papa, L. A. A. Meira, How far do we get using machine learning black-boxes?, *International Journal of Pattern Recognition and Artificial Intelligence* 26 (2012) 1261001–(1–23). doi:10.1142/S0218001412610010.
- [2] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, *ACM Computing Surveys* 51 (2018) 1–42. doi:10.1145/3236009.
- [3] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, *Nature Machine Intelligence* 1 (2019) 206–215. doi:10.1038/s42256-019-0048-x.
- [4] E. M. Kenny, C. Ford, M. Quinn, M. T. Keane, Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies, *Artificial Intelligence* 294 (2021) 103459. doi:10.1016/j.artint.2021.103459.
- [5] B. Baesens, R. Setiono, C. Mues, J. Vanthienen, Using neural network rule extraction and decision tables for credit-risk evaluation, *Management Science* 49 (2003) 312–329. doi:10.1287/mnsc.49.3.312.12739.
- [6] B. Baesens, R. Setiono, V. De Lille, S. Viaene, J. Vanthienen, Building credit-risk evaluation expert systems using neural network rule extraction and decision tables, in: V. C. Storey, S. Sarkar, J. I. DeGross (Eds.), *ICIS 2001 Proceedings*, Association for Information Systems, 2001, pp. 159–168. URL: <http://aisel.aisnet.org/icis2001/20>.
- [7] M. T. A. Steiner, P. J. Steiner Neto, N. Y. Soma, T. Shimizu, J. C. Nievola, Using neural network rule extraction for credit-risk evaluation, *International Journal of Computer Science and Network Security* 6 (2006) 6–16. URL: http://paper.ijcsns.org/07_book/200605/200605A02.pdf.
- [8] L. Franco, J. L. Subirats, I. Molina, E. Alba, J. M. Jerez, Early breast cancer prognosis prediction and rule extraction using a new constructive neural network algorithm, in: *Computational and Ambient Intelligence (IWANN 2007)*, volume 4507 of *LNCS*, Springer, 2007, pp. 1004–1011. doi:0.1007/978-3-540-73007-1_121.
- [9] Y. Hayashi, R. Setiono, K. Yoshida, A comparison between two neural network rule extraction techniques for the diagnosis of hepatobiliary disorders, *Artificial intelligence in Medicine* 20 (2000) 205–216. doi:10.1016/s0933-3657(00)00064-6.
- [10] G. Bologna, C. Pellegrini, Three medical examples in neural network rule extraction, *Physica Medica* 13 (1997) 183–187. URL: <https://archive-ouverte.unige.ch/unige:121360>.
- [11] R. Setiono, B. Baesens, C. Mues, Rule extraction from minimal neural networks for credit card screening, *International Journal of Neural Systems* 21 (2011) 265–276. doi:10.1142/S0129065711002821.
- [12] A. Hofmann, C. Schmitz, B. Sick, Rule extraction from neural networks for intrusion detection in computer networks, in: *2003 IEEE International Conference on Systems, Man and Cybernetics*, volume 2, IEEE, 2003, pp. 1259–1265. doi:10.1109/ICSMC.2003.1244584.
- [13] A. Azcarraga, M. D. Liu, R. Setiono, Keyword extraction using backpropagation neural networks and rule extraction, in: *The 2012 International Joint Conference on Neural Networks (IJCNN 2012)*, IEEE, 2012, pp. 1–7. doi:10.1109/IJCNN.2012.6252618.

- [14] J. W. Lloyd (Ed.), *Computational Logic*, Springer, 1990. doi:10.1007/978-3-642-76274-1.
- [15] G. Metakides, A. Nerode, Principles of logic and logic programming, volume 13 of *Studies in Computer Science and Artificial Intelligence*, Elsevier, 1996. URL: <https://www.elsevier.com/books/principles-of-logic-and-logic-programming/metakides/978-0-444-81644-3>.
- [16] A. Colmerauer, P. Roussel, The birth of Prolog, in: *History of programming languages—II*, ACM, 1996, pp. 331–367. doi:10.1145/234286.1057820.
- [17] G. Ciatto, D. Calvaresi, M. I. Schumacher, A. Omicini, An abstract framework for agent-based explanations in AI, in: A. El Fallah Seghrouchni, G. Sukthankar, B. An, N. Yorke-Smith (Eds.), *19th International Conference on Autonomous Agents and MultiAgent Systems, IFAAMAS, 2020*, pp. 1816–1818. URL: <http://ifaamas.org/Proceedings/aamas2020/pdfs/p1816.pdf>.
- [18] Z. C. Lipton, The mythos of model interpretability, *Queue* 16 (2018) 31–57. doi:10.1145/3236386.3241340.
- [19] M. W. Craven, J. W. Shavlik, Using sampling and queries to extract rules from trained neural networks, in: *Machine Learning Proceedings 1994*, Elsevier, 1994, pp. 37–45. doi:10.1016/B978-1-55860-335-6.50013-1.
- [20] J. Huysmans, B. Baesens, J. Vanthienen, ITER: An algorithm for predictive regression rule extraction, in: *Data Warehousing and Knowledge Discovery (DaWaK 2006)*, Springer, 2006, pp. 270–279. doi:10.1007/11823728_26.
- [21] F. Sabbatini, G. Ciatto, A. Omicini, GridEx: An algorithm for knowledge extraction from black-box regressors, in: D. Calvaresi, A. Najjar, M. Winikoff, K. Främling (Eds.), *Explainable and Transparent AI and Multi-Agent Systems. Third International Workshop, EXTRAAMAS 2021, Virtual Event, May 3–7, 2021, Revised Selected Papers*, volume 12688 of *LNCS*, Springer Nature, Basel, Switzerland, 2021, pp. 18–38. doi:10.1007/978-3-030-82017-6_2.
- [22] M. W. Craven, J. W. Shavlik, Extracting tree-structured representations of trained networks, in: D. S. Touretzky, M. C. Mozer, M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, The MIT Press, 1996, pp. 24–30. URL: <http://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks.pdf>.
- [23] G. Ciatto, R. Calegari, A. Omicini, D. Calvaresi, Towards XMAS: eXplainability through Multi-Agent Systems, in: C. Savaglio, G. Fortino, G. Ciatto, A. Omicini (Eds.), *AI&IoT 2019 – Artificial Intelligence and Internet of Things 2019*, volume 2502 of *CEUR Workshop Proceedings*, CEUR WS, 2019, pp. 40–53. URL: <http://ceur-ws.org/Vol-2502/paper3.pdf>.
- [24] R. Calegari, G. Ciatto, A. Omicini, On the integration of symbolic and sub-symbolic techniques for XAI: A survey, *Intelligenza Artificiale* 14 (2020) 7–32. doi:10.3233/IA-190036.
- [25] N. Barakat, J. Diederich, Eclectic rule-extraction from support vector machines, *International Journal of Computer and Information Engineering* 2 (2008) 1672–1675. doi:10.5281/zenodo.1055511.
- [26] D. Martens, B. Baesens, T. Van Gestel, J. Vanthienen, Comprehensible credit scoring models using rule extraction from support vector machines, *European Journal of Operational Research* 183 (2007) 1466–1476. doi:10.1016/j.ejor.2006.04.051.
- [27] R. Setiono, W. K. Leow, J. M. Zurada, Extraction of rules from artificial neural networks

- for nonlinear regression, *IEEE Transactions on Neural Networks* 13 (2002) 564–577. doi:10.1109/TNN.2002.1000125.
- [28] G. P. J. Schmitz, C. Aldrich, F. S. Gouws, ANN-DT: an algorithm for extraction of decision trees from artificial neural networks, *IEEE Transactions on Neural Networks* 10 (1999) 1392–1401. doi:10.1109/72.809084.
- [29] K. Saito, R. Nakano, Extracting regression rules from neural networks, *Neural Networks* 15 (2002) 1279–1288. doi:10.1016/S0893-6080(02)00089-8.
- [30] R. Konig, U. Johansson, L. Niklasson, G-REX: A versatile framework for evolutionary data mining, in: 2008 IEEE International Conference on Data Mining Workshops (ICDM 2008 Workshops), 2008, pp. 971–974. doi:10.1109/ICDMW.2008.117.
- [31] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, *Classification and Regression Trees*, CRC Press, 1984.
- [32] A. A. Freitas, Comprehensible classification models: a position paper, *ACM SIGKDD Explorations Newsletter* 15 (2014) 1–10. doi:10.1145/2594473.2594475.
- [33] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, B. Baesens, An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models, *Decision Support Systems* 51 (2011) 141–154. doi:10.1016/j.dss.2010.12.003.
- [34] P. M. Murphy, M. J. Pazzani, Id2-of-3: Constructive induction of m-of-n concepts for discriminators in decision trees, in: *Machine Learning Proceedings 1991*, Elsevier, 1991, pp. 183–187.
- [35] J. R. Quinlan, *C4.5: Programming for machine learning*, Morgan Kaufmann (1993). URL: <https://dl.acm.org/doi/10.5555/152181>.
- [36] J. R. Quinlan, Simplifying decision trees, *International Journal of Man-Machine Studies* 27 (1987) 221–234. doi:10.1016/S0020-7373(87)80053-6.
- [37] R. Andrews, J. Diederich, A. B. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, *Knowledge-Based Systems* 8 (1995) 373–389. doi:10.1016/0950-7051(96)81920-4.
- [38] G. G. Towell, J. W. Shavlik, Extracting refined rules from knowledge-based neural networks, *Machine Learning* 13 (1993) 71–101. doi:10.1007/BF00993103.
- [39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. VanderPlas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research (JMLR)* 12 (2011) 2825–2830. URL: <https://dl.acm.org/doi/10.5555/1953048.2078195>.
- [40] G. Ciatto, R. Calegari, E. Siboni, E. Denti, A. Omicini, 2P-КТ: logic programming with objects & functions in Kotlin, in: R. Calegari, G. Ciatto, E. Denti, A. Omicini, G. Sartor (Eds.), *WOA 2020 – 21th Workshop “From Objects to Agents”*, volume 2706 of *CEUR Workshop Proceedings*, CEUR WS, 2020, pp. 219–236. URL: <http://ceur-ws.org/Vol-2706/paper14.pdf>, 21st Workshop “From Objects to Agents” (WOA 2020), Bologna, Italy, 14–16 September 2020. Proceedings.
- [41] G. Ciatto, R. Calegari, A. Omicini, Lazy stream manipulation in Prolog via backtracking: The case of 2P-КТ, in: W. Faber, G. Friedrich, M. Gebser, M. Morak (Eds.), *Logics in Artificial Intelligence*, volume 12678 of *LNCS*, Springer, 2021, pp. 407–420. doi:10.1007/978-3-030-75775-5_27, 17th European Conference, JELIA 2021, Virtual Event, May

- 17–20, 2021, Proceedings.
- [42] J. Dougherty, R. Kohavi, M. Sahami, Supervised and unsupervised discretization of continuous features, in: A. Prieditis, S. J. Russell (Eds.), *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning*, Tahoe City, California, USA, July 9–12, 1995, Morgan Kaufmann, 1995, pp. 194–202. doi:10.1016/b978-1-55860-377-6.50032-3.
- [43] Y. Yang, G. I. Webb, X. Wu, Discretization methods, in: O. Maimon, L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook*, 2nd ed, Springer, 2010, pp. 101–116. doi:10.1007/978-0-387-09823-4_6.
- [44] R. Kerber, Chimerge: Discretization of numeric attributes, in: W. R. Swartout (Ed.), *Proceedings of the 10th National Conference on Artificial Intelligence*, San Jose, CA, USA, July 12–16, 1992, AAAI Press / The MIT Press, 1992, pp. 123–128. URL: <http://www.aaai.org/Library/AAAI/1992/aaai92-019.php>.
- [45] K. M. Ho, P. D. Scott, Zeta: A global method for discretization of continuous variables, in: D. Heckerman, H. Mannila, D. Pregibon (Eds.), *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, Newport Beach, California, USA, August 14–17, 1997, AAAI Press, 1997, pp. 191–194. URL: <http://www.aaai.org/Library/KDD/1997/kdd97-037.php>.
- [46] M. Boullé, Khiops: A statistical discretization method of continuous attributes, *Machine Learning* 55 (2004) 53–69. doi:10.1023/B:MACH.0000019804.29836.05.
- [47] L. A. Kurgan, K. J. Cios, CAIM discretization algorithm, *IEEE Trans. Knowl. Data Eng.* 16 (2004) 145–153. doi:10.1109/TKDE.2004.1269594.
- [48] A. Cano, D. T. Nguyen, S. Ventura, K. J. Cios, ur-caim: improved CAIM discretization for unbalanced and balanced data, *Soft Comput.* 20 (2016) 173–188. doi:10.1007/s00500-014-1488-1.
- [49] H. Elhilbawi, S. Eldawlatly, H. Mahdi, The importance of discretization methods in machine learning applications: A case study of predicting ICU mortality, in: A. E. Hassanien, K. Chang, M. Tang (Eds.), *Advanced Machine Learning Technologies and Applications - Proceedings of AMLTA 2021*, Cairo, Egypt, March 22–24, 2021, volume 1339 of *Advances in Intelligent Systems and Computing*, Springer, 2021, pp. 214–224. doi:10.1007/978-3-030-69717-4_23.
- [50] M. Gavanelli, F. Rossi, Constraint logic programming, in: A. Dovier, E. Pontelli (Eds.), *A 25-Year Perspective on Logic Programming: Achievements of the Italian Association for Logic Programming*, GULP, volume 6125 of *LNCS*, Springer, 2010, pp. 64–86. doi:10.1007/978-3-642-14309-0_4.
- [51] J. Jaffar, M. J. Maher, Constraint logic programming: A survey, *J. Log. Program.* 19/20 (1994) 503–581. doi:10.1016/0743-1066(94)90033-7.
- [52] D. Calvaresi, G. Ciatto, A. Najjar, R. Aydoğan, L. Van der Torre, A. Omicini, M. Schumacher, EXPECTATION: Personalized explainable artificial intelligence for decentralized agents with heterogeneous knowledge, in: D. Calvaresi, A. Najjar, M. Winikoff, K. Främling (Eds.), *Explainable and Transparent AI and Multi-Agent Systems. Third International Workshop, EXTRAAMAS 2021*, Virtual Event, May 3–7, 2021, Revised Selected Papers, volume 12688 of *LNCS*, Springer Nature, Basel, Switzerland, 2021, pp. 331–343. URL: http://link.springer.com/10.1007/978-3-030-82017-6_20. doi:10.1007/978-3-030-82017-6_20.

Investigating Adjustable Social Autonomy in Human Robot Interaction

Filippo Cantucci, Rino Falcone and Cristiano Castelfranchi

Institute of Cognitive Science and Technology, National Research Council of Italy, (ISTC-CNR), Rome

Abstract

More and more often, Human Robot Interaction(HRI) applications require the design of robotics systems whose decision process implies the capability to evaluate not only the physical environment, but especially the mental states and the features of its human interlocutor, in order to adapt their *social autonomy* every time humans require the robot's help. Robots will be really cooperative and effective when they will expose the capability to consider not only the goals or interests explicitly required by humans, but also those one that are not declared and to provide help that go beyond the literal task execution. In order to improve the quality of this kind of smart help, a robot has to operate a meta-evaluation of its own predictive skills to build a model of the interlocutor and of her/his goals. The robot's capability to *self-trust* its skills to interpret the interlocutor and the context, is a fundamental requirement for producing smart and effective decisions towards humans. In this work we propose a simulated experiment, designed with the goal to test a cognitive architecture for trustworthy human robot collaboration. The experiment has been designed in order to demonstrate how the robot's capability to learn its own level of self-trust on its predictive abilities in perceiving the user and building a model of her/him, allows it to establish a trustworthy collaboration and to maintain an high level of user's satisfaction, with respect to the robot's performance, also when these abilities progressively degrade.

Keywords

Trustworthy HRI, Robot Autonomy Adaptation, Theory of Mind, Transparency, Cognitive Modelling

1. Introduction

In today's world, artificial intelligence systems are playing a crucial role in our daily lives. The decisions made by machines are leaving a profound impact on our society and are involving almost every aspect of our life. Different kinds of artificial systems, whose behaviours is based on statistical tools, AI algorithms, machine learning models are used in applications such as healthcare, government, business, judicial and political spheres. Decisions made by AI systems lead to beat some of the best human player [1], to make super accurate medical diagnostics [2], to help companies in customers support [3] and so on. These decisions are more oriented to superhuman computations and performances, than brain-inspired or psychological paradigms. With the enormous impact that AI systems have in society, it is crucial to assure that all these systems we are relying on are trustworthy. Trustworthy AI is largely considered one of the topics much more demanding in the artificial intelligence field, not only in research, but also in institutions [4, 5], due to the huge impact that AI systems are having in society.

WOA 2021: Workshop "From Objects to Agents", September 1–3, 2021, Bologna, Italy

✉ filippo.cantucci@istc.cnr.it (F. Cantucci); rino.falcone@istc.cnr.it (R. Falcone);

cristiano.castelfranchi@istc.cnr.it (C. Castelfranchi)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

As mentioned above, AI moved from human psychology inspired models (i.e. decision trees in expert systems) to deep neural networks, machine learning, Big Data and so on. If this type of approach proved to be very powerful in computational and performance terms, it increased the gap between super intelligent agents and humans, in terms of trustworthy cooperation between humans and artificial systems. We do not consider just the cases in which results provided by artificial systems have been extremely dangerous for humans [6, 7] (trustworthiness as *accuracy, robustness, non-discrimination, privacy, security*; we focus on those dimensions of trustworthiness (e.g. adaptation to human *autonomy*, behavior *transparency* and *explainability*) that are involved when humans and artificial systems, in particular *robots*, have to interact [8, 9] and cooperate [10] with each other, and humans have to establish a deep relationship of *trust* [11, 12] every time they include robots as part of their plans or goals (*task delegation* and *adoption* [13]). Trust is not just the result of the frequency with which an agent produces the desired behavior or result; trust is a much more complex attitude, including a causal attribution, an estimation, an ascription of several internal factors that play a causal role in the activation and control of the behavior [14].

1.1. intelligent cooperation

Cooperation is based on different and complementary kinds of attitudes and reasons from the partners involved. Let's consider the following collaborative scenario: a human X (the *trustor*) and a robot Y (the *trustee*) collaborate so that X has to trust Y , in a specific context, for executing a task τ and realizing the results that include or correspond to the X 's $Goal_X(g) = g_X$ [14]. In this context, X relies on Y for realizing some part of the task she/he has in mind (*task delegation*); on its side, Y decides to help X , to replace her/him and perform a sequence of actions that are included in the X 's plan, in order to achieve some of her/his goals or sub-goals (*task adoption*). The capability to implement a smart task adoption distinguishes a collaborator from a simple tool, and presupposes intelligence and *autonomy* [15]. Being truly cooperative implies more than the simple concept of *execution* of a prescribed action. For example, in order to adopt some goal of X in an intelligent form, Y has to understand the X 's mental states (i.e. goals, beliefs, expectations about Y 's behavior) and it has to adjust the delegated action to the represented mental states, to the context and to its own current abilities and characteristics. In their much complex sense, cooperation and help require more autonomy and *initiative*. A real collaborative trustee should provide to the trustor different kind of help, according with [15]:

- *Sub help*: Y satisfies a sub-part of the delegated world-state (so satisfying just a sub-goal of X),
- *Literal help*: Y adopts exactly what has been delegated by X ,
- *Over help*: Y goes beyond what has been delegated by X without changing X 's plan (but including it within a hierarchically superior plan),
- *Critical-Over help*: Y realizes an over help and in addition modifies also the original plan/action (included in the new meta-plan),
- *Critical help*: Y satisfies the relevant results of the requested plan/action (the goal), but modifies that plan/action,
- *Critical-Sub help*: Y realizes a sub help and in addition modifies the (sub) plan/action,

- *Hyper-critical help*: Y adopts goals or interests of X that X itself did not take into account (at least, in that specific interaction with Y): by doing so, Y neither performs the specific delegated action/plan nor satisfies the results that were delegated. In practice, Y satisfies other goals/interests of X by realizing a new plan/action.

Y has to exploit its autonomy, competence and cognitive skills to find the better or a possible solution for X's goal. This not necessarily should require a negotiation, discussion, agreement; it might be an initiative of Y by expecting that X will understand why. This is precisely what intelligent robots must have and these are the kind of partners the humans need.

How would this advanced form of cooperation would be possible? What are some of the capabilities that a robot has to show for enhancing trust in its human interlocutor? A smart and trust-based collaboration between humans and intelligent robots requires, among many others things, complex cognitive capabilities these artificial systems must be endowed with: mental attribution, adjustable autonomy, user profiling and user behavior adaptation, behavior transparency. Besides the capabilities to evaluate the interlocutor and/or the contextual physical environment, a robot (as a trustee) should be able also to operate a *meta-evaluation*: how much itself would be able to interpret and produce the evaluations regarding the trustor? How much is reliable its capability to perceive or infer the trustor's features? On the basis of its own capabilities to perceive or to act in the world, the hypothesis or prediction it has made, the chosen course of action, are the best or the most effective, with respect to the needs, the features and the mental states of the interlocutor? Smart help has to be based on different capabilities to interpret the environment and the interacting user, but first of all, it has to be based on the robot's capability to realistically self-assess the level of trustworthiness on its ability to interpret the collaborative and potentially uncertain context, including the interacting user [16, 17]. The outcome of the meta-evaluation expressed above represents the *robot's self-trust* for adopting a delegated task. In practice, the robot uses this evaluation of its own specific abilities as a filter for their use with respect to the interlocutors with whom it is interacting. The robot learns the trustworthiness of its skills and, on the basis of the context and the task to carry out, establishes which skills to use and how trustworthy (from its point of view) will be the solution it will propose to its interlocutor. So robot's self-trust can be viewed as a precondition for exploiting the robot's interpretative skills accordingly to its own interlocutor, in order to foster a true and deep relationship of collaboration and trust with her/him.

1.2. the risk of collaborative conflicts

A form of intelligent help that can provide results beyond those explicitly requested by the interlocutor implies risks. One of the possible consequences of this form of help can be the emergence of *collaborative conflicts* between the human (the trustor) and the robot (the trustee) that adopts the task, due to the robot's willingness to collaborate and to help the user better and more deeply than required. Sometimes, the difference between the results of the adopted task provided by the robot and the user's expectations, could lead the interlocutor to a complete lack of trust towards the robot. We are not just considering the robot's failure in the precise delegated task: failures become more evident every time the robot goes beyond the delegated task and the results are too much distant (or even in conflict) from the user's expectations. Among

humans these conflicts can be mitigated by the experience: humans learn to measure their competence in achieving specific results, or making the right prediction about the correctness of a chosen behaviour, on the basis of the context and the interlocutor; furthermore, on this basis, they learn to self-trust their own abilities/skills (with respect to both the interlocutors and the tasks). Similarly, robots can learn to trust their capabilities to evaluate the interlocutors (and consequently to build and use the cognitive models they attribute to them) through a repetitive interactions with humans. For example, a robot can exploit the feedback provided by its interlocutor any time she/he delegates to it a task and receive an evaluation (i.e. user's satisfaction) on the results of the robot's adoption process.

1.3. our contribution

In this work we propose a preliminary, simulated experiment, designed with the goal to test a cognitive architecture [18] for trustworthy human robot collaboration. A complete description of both the cognitive architecture and experiment are reported in [19]. The designed architecture allows a BDI robot [20], with its own mental states (beliefs, goals, plans and so on) to expose a wide range of cognitive skills that support an effective, smart and trustworthy collaboration, every time a human user delegates to it a task to achieve in her/his place. In particular, we focused on endowing the robot with the capabilities to i) adapt its level of collaborative autonomy, providing an intelligent help (based on the levels of help formalized in [15]) every time the user delegates to it a task to accomplish; the autonomy adaptation leverages on the agent's capabilities to profile the user and to have a theory of mind of her/him [21] ii) learn its limits in interpreting the needs of the interlocutor, by measuring its degree of self-trust on its predictive abilities in perceiving the user; the agent chooses those abilities that maximize the user's task performance evaluation. In particular the simulation aims at demonstrating how the robot's capability to learn the level of self-trust on its predictive abilities in perceiving the user, allows it to choose the best user's model (as a collection of mental states) and to preserve an high level of the user's task performance evaluation.

2. The proposed experiment

The experiment designed for testing the cognitive architecture proposed in [18], has been implemented by exploiting the well known multi-agent oriented programming (MAOP) framework *JaCaMo* [22], that integrates three different multi-agent programming levels: agent-oriented (AOP), environment-oriented (EOP) and organization-oriented programming (OOP). Basically, the experiment simulates the process of task delegation and task adoption between a robot and multiple users, grouped in classes of users, in a specific application domain.

2.1. the experimental settings

We figured the following interactive scenario: the robot is a touristic assistant that helps people to organize different touristic activities offered by a city (i.e. eat in a restaurant, visit a museum, visit a monument, drink something in a bar, enjoy the city doing multiple daily activities). The experiment is based on the interaction between two agents: the user and the robot. Both of

them are implemented as Jason [23] agents. The user has her/his own mental states represented in form of beliefs, goals and plans and interacts with the robot by delegating to it a task. On its side, the robot is able to represent and attribute mental states to the user and to itself and, on the basis of its capabilities to profile the user and build a model of her/him, to adopt the delegated task at different levels of help.

The experiment has been designed with the goal to show the importance for a robot to self estimate the level of trustworthiness associated to its expertise in building a profile of the interacting user. This capability lets the robot choose the best and suitable task to adopt with respect to the user's features, also when its skills progressively degrade and can be considered not trustworthy. Indeed, the robot is able to sort these skills on the basis of the corresponding level of trustworthiness, and leverage on the most trustworthy among them for deciding how to adopt the task delegated. As mentioned above, two agents populate the simulation: the agent robot \mathcal{R} and the agent user \mathcal{U} . The agent \mathcal{U} is characterized by a profile $\mathcal{P}_{\mathcal{U}} = \{Age, Economic\ status, Category, Education\ level, Company\}$, a collection of five physical and social features. Every feature is associated to *sub-components* and real values $r_{H_i} \in [0, 1]$ belonging to specific intervals that are bonded to the sub-components. Table 1 shows the relations between features, sub-components and intervals. We decide to consider these groups of user's demographic features, because they are all concrete characteristics that help the robot, operating in a touristic domain, to narrow down which segment of population the interacting users best fit into. That means the robot can split a larger group into subgroups based on, for example, their educational level, age, income. This kind of physical, social and relational features are largely used, easy to collect and they are reasonably good predictors of user preferences [24]. For example, demographic recommendation system generate recommendations based on the user demographic attributes [25, 26]. In our case the robot is able to filter and categorize the interacting users based on their attributes and recommends the most suitable service (restaurant, museum, monument or bar) by utilizing the chosen demographic data collected in its profile. The partition of the features into sub components is an approximation that allows the robot to cluster users into a series of discrete categories, commonly used by human for identify expected behaviors or character traits, related to that particular category [27].

Users are organized into *classes of populations*: each class collects together users with the same profile (in terms of sub-components). Each user of a class distinguishes from the others due to five real values r_{H_i} for $i = 1, \dots, 5$ randomly picked up from the interval associated to the sub-components. The decision making system of \mathcal{R} is designed following the principles described in [18]. The robot is able to recognize and classify, as set of specific sub-components, the features collected in $\mathcal{P}_{\mathcal{U}}$, consistent with the table 1. \mathcal{R} is not always able to infer all the features of \mathcal{U} ; that depends on the robot's *accuracy* to estimate a feature of $\mathcal{P}_{\mathcal{U}}$. In this experiment we decide to define two levels of accuracy: a low level of accuracy, that means the robot has great difficulties in distinguishing a feature, and an high level of accuracy, corresponding to the fact that it is perfectly able to recognize a feature. We have designed the simulation so that \mathcal{R} can estimate the sub-components collected in $\mathcal{P}_{\mathcal{U}}$, but it is not able to perfectly recognize the real values r_{H_i} for each user; because of that, it associates to every feature it has estimated, the mean value of the corresponding intervals defined in the table 1. We observe that, if the robot profiles a feature correctly, the corresponding mean value will be close to the value r_{H_i} of the user (for that feature), while if the robot is not able to infer correctly the feature, this value will

Feature	Sub-component [interval]
Age	young [0, 0.33] adult [0.34, 0.66] old [0.67, 1]
Category	loco tourist [0, 0.33] foreign tourist [0.34, 0.66] resident [0.67, 1]
Economic status	low economic status [0, 0.33] medium economic status [0.34, 0.66] high economic status [0.67, 1]
Education level	low education [0, 0.33] medium education [0.34, 0.66] high education [0.67, 1]
Company	single [0, 0.33] in couple [0.34, 0.66] in family [0.67, 1]

Table 1
Map of the relations between features, sub-components and intervals

be distant from that of the user.

It is important to specify that the robot’s beliefs are organized according to the features that are classified within \mathcal{P}_U and which are perceivable by the robot itself. \mathcal{R} has available (among the set of its mental states) a subset of beliefs where are represented information about a finite number of services that a city offers: restaurants, museums, monuments to visit and places for having fun (night clubs, bar and so on). Each service is described with respect to the features described in table 1: for example, in the robot’s beliefs base exist restaurants much more suitable to young people, instead of monuments or museums much more adapt to people with an high level of education, and so on. The robot is able to select the most suitable service with respect to the features that it has been able to infer from \mathcal{U} . This criterion of choice can lead the robot to select the most adapt service with respect to the user’s profile or not, on the basis of its own profiling skills accuracy.

2.2. the experiment description

The experiment is a simulation of several *trials* – interactions between \mathcal{R} and 100 users belonging to the same class (population of users) – involving the robot and different users. Every interaction reproduces the mechanism of delegation and adoption: \mathcal{U} delegates a task to \mathcal{R} and the robot adopts the task at different levels of intelligent help, among those introduced in section 1. We defined a class of population \mathcal{C}_1 formed by users that have the following profile (collection of sub-components): $\mathcal{P}_U = \{young, medium\ Economic\ Status, foreign\ Tourist, medium\ Education, single\}$. Each interaction requires that the current user delegates to the robot the goal to *eat in a restaurant*. The request might be further specified by giving the name of the restaurant, the type of restaurant and the area of the city in which it is located. We decide to specify only the area of the city where the user desires to eat.

2.3. building robot’s self-trust

The robot \mathcal{R} builds its self-trust for adopting the delegated task τ by means of a training phase, with the goal to learn the levels of trustworthiness associated to its own user profiling capabilities. The training phase requires that the robot performs an interaction with a population of a specific class formed by 100 users. Every user \mathcal{U} delegates to \mathcal{R} the same task (i.e. eat in a restaurant); for its part, the robot adopts the task at a literal level of help. At every interaction \mathcal{R} computes a *robot’s skill trustworthiness* value, each for every feature that forms $\mathcal{P}_{\mathcal{U}}$. These values depend on the feedback provided by the users during the training phase. We designed a robot that explicitly asks for feedback, once it accomplishes a task to be achieved on behalf of \mathcal{U} . Every question the robot asks humans aims at evaluating how the delegating user has been satisfied by the robot’s task adoption; different user’s satisfaction dimensions are investigated, each of them corresponding with the different abilities of the robot to profile the user. In this way \mathcal{R} can evaluate how each of its skills performs (and to measure its trustworthiness) with respect to build $\mathcal{P}_{\mathcal{U}}$. Furthermore, \mathcal{R} can sort the skills on the basis of the measured level of trustworthiness.

2.4. The user’s satisfaction function

We have introduced a *user’s satisfaction* function $S_{\mathcal{U}}$ that computes the global user’s satisfaction regarding the collaboration offered by the robot; the robot aims at maximizing this function every time it interacts with a new user. $S_{\mathcal{U}}$ is the linear combination between a term P_{τ} that measures how much the user has been satisfied by the results of \mathcal{R} in performing precisely the delegated task and a term $S_{\mathcal{U}plus}$ that measures how much the user has been satisfied by the additional, not explicitly required part of the plan performed by the robot in its smart collaboration. Both terms are affected by the robot’s capabilities to profile the user and to learn their corresponding trustworthiness. In particular, \mathcal{R} ’s profiling capability is quantified by calculating how the robot has adapted the task to the real user’s features that form $\mathcal{P}_{\mathcal{U}}$: the greater is this measure for each feature, the more accurate is the robot’s capability to profile the user on that feature and the greater are the user’s satisfaction components mentioned above. As will be clear in the results section (section 3), both components P_{τ} and $S_{\mathcal{U}plus}$ are designed so that they vary in the codomain $[0, 1]$, while $S_{\mathcal{U}}$ varies in the codomain $[-1, 2]$.

2.5. the experiment’s phases

The experiment is structured as follows:

1. the robot implements a first trial with a population of class \mathcal{C}_1 . During this multiple interaction, the robot decides to adopt the task at the level of help it considers appropriate to the user and the context. The phase is designed so that \mathcal{R} infers the feature *category* with a low level of accuracy, while the other features of $\mathcal{P}_{\mathcal{U}}$ are inferred with an high level of accuracy;
2. the robot implements a second trial with the same population of class \mathcal{C}_1 exploited in the previous phase. During the trial, the robot decides to adopt the task at the level of help it considers appropriate to the user and the context. In this case \mathcal{R} ’s capability to infer

\mathcal{P}_U degrade because the features *age*, *category*, *education* are affected by a low level of accuracy (features *company* and *economic status* are still inferred with an high level of accuracy);

3. the robot starts a training phase with a new population of class \mathcal{C}_1 , in order to learn its own level of self-trust. In this phase, \mathcal{R} has the same profiling skills described at point 1. Please recall that, during the training phase, \mathcal{R} adopts the task at a literal level of help;
4. the robot starts a second training phase with a new population of class \mathcal{C}_1 , but this time its profiling skills are the same described at point 2;
5. the trials described at points 1 and 2 are repeated, but this time the robot exploits what it has learned respectively in the context described at the point 3 and 4, in order to achieve the task adoption process.

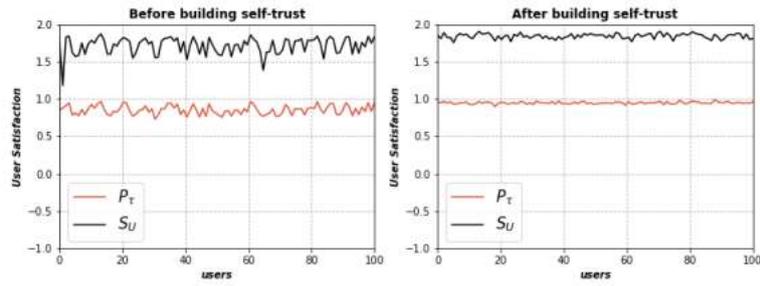
3. Results

In this section we present the results of the experiment designed in order to address the research purpose previously defined: demonstrate how building robot’s self-trust is a precondition for providing smart and trustworthy collaboration, every time a user requires the robot’s help. The plots shown in Figure 1 compares the results obtained after the execution of each experiment’s phase described in section 2.5.

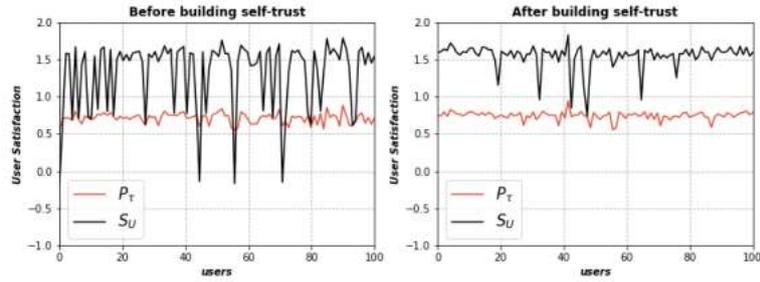
Let’s start by describing the Figure 1a. This plots refer to the case when the robot’s capability to recognize the feature *category* is inaccurate, while are accurate the capability to recognize the remaining features collected in \mathcal{P}_U . The left plots show the distribution of P_τ and S_U obtained when \mathcal{R} performs a trial with a population of class 1 and it is not yet able to evaluate the level of trustworthiness of its profiling skills. Instead, the right chart shows P_τ and S_U trends when the robot’s capability are the same described in section 2.5 at point 1, but it has learned to self evaluate the trustworthiness of its own profiling skills.

Figure 1b displays the trends of the user’s satisfaction function S_U and its component P_τ in case the robot performs a trial with a population of class 1 and its profiling skills are such that it cannot correctly recognize the features *age,category,education*, while it infers the user’s *economic status* and *company* with an high accuracy (point 2 described in section 2.5). In particular, the left part of the figure shows the results in case the robot is not able to self evaluate the trustworthiness of its profiling skills, while the right part shows how the user’s satisfaction change once the robot has learned to attribute a specific level of trustworthiness to its profiling skills.

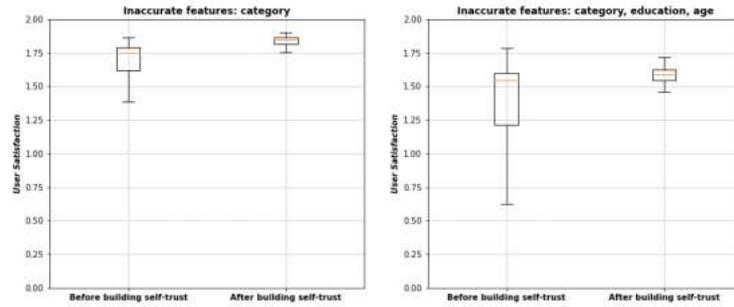
Finally, Figure 1c shows the box plots comparing the distributional characteristics of S_U before and after the robot’s self-trust building process. In particular, the left box plot and the right box plot refer to the cases of the robot is capable to profile the user with the conditions described respectively at point 1 and 2 of the section 2.5. Comparing the plots in Figure 1a we observe how the robot’s capability to recognize the level of trustworthiness of its profiling skills is crucial for maintaining an high level of the user’s satisfaction about the robot’s performance. This capability become more important when the robot decides to adopt the delegated task to a level of help different with respect the literal one. Indeed, despite the robot provides unexpected results to the user, its own capabilities to adapt these results by leveraging on the capabilities



(a)



(b)



(c)

Figure 1: Figure 1a and 1b show the trend of the curves representing the user's satisfaction, obtained after each phase described in section 2.5: each plot represents the trend of the component P_τ (light red line) and the trend of S_U (dark red line) as combination of P_τ and $S_{U_{plus}}$. Figure 1c shows a statistical description of the impact of the self-trust building process in the level of user's satisfaction on the robot's smart collaboration.

that it considers trustworthy, allows the robot to provide unexpected but suitable results, that are appropriate to the user himself/herself. The plots in Figure 1a and the left box plot of Figure 1c, show how the mean (and the median) value of S_U increases after the robot has learned its self-trust level; moreover, the spread and the skewness of the S_U distribution is drastically reduced by the robot's capability to self evaluate the trustworthiness of its profiling skills. Figure 1b and the right box plot of Figure 1c show the benefits of the building self-trust process on

the user task performance evaluation. In this case, the increase of the median value of S_U is less evident than for the previous case analyzed, but the training phase impact remains evident on the spread and the skewness of the distribution. This means that, also when the robot's profiling skills degrade, its capability to evaluate their trustworthiness continues to allow the robot to provide unexpected but suitable results with respect to the needs of the users. It is also relevant to underline how the effective performance of the robot's help depends on the width and variety of the database of the accessible services with respect to the selected features. In fact, with a very low number of trustworthy features (given the low level of accuracy of three of them) the result of the adoption could be really very good only if the database contains services responding, with very high performance, to the two remaining features independently to the values of the three (degraded) features.

4. Final Remarks

Cooperation is one of the main social activities exploited by humans for gaining resources, in terms of goals achieved, shared knowledge and so on. The increasing intelligent technology surrounding us is becoming crucial for our own social development, and, as a consequence, the need of trusting these supporting and sophisticated tools is becoming every day more stringent. But, if on the one hand these systems are becoming more intelligent and sophisticated, on the other hand they show a strong lack in the ability to collaborate effectively with humans. Despite the complexity of the problem they can solve, they continue to have just a passive supporting role in the collaboration with humans. For being not only executive tools, these intelligent systems (i.e. robots, chat-bots, autonomous cars and so on) should expose the capability to behave in a critical way with respect to the needs/goals of their interacting users. Indeed, the collaboration becomes deep and effective when a system is able to provide not declared, unexpected results but compatible with the context, the needs of the user and the capabilities of the system itself. The level of autonomy of robots or other artificial agents, it should be such that such systems can exercise a certain level of discretion in achieving the task delegated but humans. But, in order to foster trust in humans, they should behave having the capability to create a complex theory of mind of the interlocutors and a strong capability to self assess their own capability to carry out a task, also at a different level of help than required.

In this work we have presented the first of a series of experiments drawn for testing different aspects of a designed cognitive architecture. This architecture, based on consolidated theoretical principles (theory of adoption and delegation, theory of mind, theory of social adjustable autonomy, theory of trust) has the main goal to build robots that provide smart, trustworthy and transparent collaboration, every time a human requires their help. With this experiment we wanted to test the robustness of the designed architecture to rely on the robot's ability to learn the limits in interpreting the needs of its interlocutor, by measuring the trustworthiness of its predictive abilities. In fact, the architecture gives to a robot the capability to profile the user and to leverage on its profiling skills in an adaptive manner, by exploiting those skills that maximize the user's task performance evaluation; it allows the robot to reason about the mental states of the user (beliefs, goals, plans and intentions) and makes it capable to modulate its autonomy for achieving the delegated task. One of the main problems in intelligent collaboration between

humans is the possibility of misunderstandings that can lead to conflicts between cooperators. We call these collaborative conflicts, as they are based on the desire to collaborate beyond what is required but in doing this errors and discrepancies can occur. Just to minimize these conflicts and increase the robot's trustworthiness, an important requirement to introduce is the the capability of the robot itself to self trust his capabilities to build a complex model of the user. The data analyzed have shown how the process to learn the trustworthiness of its own profiling skills can lead the robot to have an effective collaboration, based not only on the actions/tasks prescribed by the user, but especially on the non declared needs and goals of the user himself/herself. Our main future work will be to move the experiment in a real environment, with a real robotic platform and real users. We will exploit the humanoid robot *Nao*, widely used in HRI applications. Furthermore, we will continue to provide simple but effective experiments that allow us to investigate different aspects of the concept of intelligent and trustworthy collaboration between robots and humans, that consider robots as cognitive agents able to interact with humans as humans do when they interact with each others.

References

- [1] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al., Mastering atari, go, chess and shogi by planning with a learned model, *Nature* 588 (2020) 604–609.
- [2] H. Fujita, Ai-based computer-aided diagnosis (ai-cad): the latest review to read first, *Radiological physics and technology* 13 (2020) 6–19.
- [3] M. Hardalov, I. Koychev, P. Nakov, Towards automated customer support, in: *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, Springer, 2018, pp. 48–59.
- [4] E. Commission, White paper on artificial intelligence-a european approach to excellence and trust, *Com (2020) 65 Final (2020)*.
- [5] N. A. Smuha, The eu approach to ethics guidelines for trustworthy artificial intelligence, *Computer Law Review International* 20 (2019) 97–106.
- [6] S. Levin, J. C. Wong, Self-driving uber kills arizona woman in first fatal crash involving pedestrian, *The Guardian* 19 (2018).
- [7] A. Schlesinger, K. P. O'Hara, A. S. Taylor, Let's talk about race: Identity, chatbots, and ai, in: *Proceedings of the 2018 chi conference on human factors in computing systems*, 2018, pp. 1–14.
- [8] G. D'Onofrio, D. Sancarlo, M. Raciti, D. Reforgiato, A. Mangiacotti, A. Russo, F. Ricciardi, A. Vitanza, F. Cantucci, V. Presutti, et al., Mario project: Experimentation in the hospital setting, in: *Italian Forum of Ambient Assisted Living*, Springer, 2017, pp. 289–303.
- [9] S. Cooper, A. Di Fava, C. Vivas, L. Marchionni, F. Ferro, Ari: The social assistive robot and companion, in: *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, IEEE, 2020, pp. 745–751.
- [10] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, S. Srinivasa, Planning with trust for human-robot collaboration, in: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 307–315.

- [11] B. C. Kok, H. Soh, Trust in robots: Challenges and opportunities, *Current Robotics Reports* (2020) 1–13.
- [12] S. Park, Multifaceted trust in tourism service robots, *Annals of Tourism Research* 81 (2020) 102888.
- [13] C. Castelfranchi, R. Falcone, Towards a theory of delegation for agent-based systems, *Robotics and Autonomous Systems* 24 (1998) 141–157.
- [14] C. Castelfranchi, R. Falcone, *Trust theory: A socio-cognitive and computational model*, volume 18, John Wiley & Sons, 2010.
- [15] C. Castelfranchi, R. Falcone, Towards a theory of delegation for agent-based systems, *Robotics and Autonomous Systems* 24 (1998) 141–157.
- [16] D. Hadfield-Menell, A. Dragan, P. Abbeel, S. Russell, The off-switch game, *arXiv preprint arXiv:1611.08219* (2016).
- [17] B. Israelsen, N. Ahmed, E. Frew, D. Lawrence, B. Argrow, Machine self-confidence in autonomous systems via meta-analysis of decision processes, in: *International Conference on Applied Human Factors and Ergonomics*, Springer, 2019, pp. 213–223.
- [18] F. Cantucci, R. Falcone, Towards trustworthiness and transparency in social human-robot interaction, in: *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*, 2020, pp. 1–6. doi:10.1109/ICHMS49158.2020.9209397.
- [19] F. Cantucci, R. Falcone, *User Modeling and User-Adapted Interaction* (Manuscript submitted for publication).
- [20] A. S. Rao, M. P. Georgeff, et al., Bdi agents: from theory to practice., in: *ICMAS*, volume 95, 1995, pp. 312–319.
- [21] S. Devin, R. Alami, An implemented theory of mind to improve human-robot shared plans execution, in: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, IEEE, 2016, pp. 319–326.
- [22] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, A. Santi, Multi-agent oriented programming with jacamo, *Science of Computer Programming* 78 (2013) 747–761.
- [23] R. H. Bordini, J. F. Hübner, Bdi agent programming in agentspeak using jason, in: *International Workshop on Computational Logic in Multi-Agent Systems*, Springer, 2005, pp. 143–164.
- [24] M. Braunhofer, M. Elahi, F. Ricci, User personality and the new user problem in a context-aware point of interest recommender system, in: *Information and communication technologies in tourism 2015*, Springer, 2015, pp. 537–549.
- [25] M. H. Mohamed, M. H. Khafagy, M. H. Ibrahim, Recommender systems challenges and solutions survey, in: *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, IEEE, 2019, pp. 149–155.
- [26] M. J. Pazzani, A framework for collaborative, content-based and demographic filtering, *Artificial intelligence review* 13 (1999) 393–408.
- [27] H. J. Swift, D. Abrams, L. Drury, R. A. Lamont, Categorization by age, *Encyclopedia of Evolutionary Psychological Science* (2018).

A Trust Model to Form Teams of Agentified AGVs in Workshop Areas

Giancarlo Fortino¹, Lidia Fotia¹, Fabrizio Messina², Domenico Rosaci³,
Giuseppe M.L. Sarnè⁴ and Claudio Savaglio⁵

¹*DIMES University of Calabria, 87036 Rende (CS), Italy*

²*DMI University of Catania, 95126 Catania, Italy*

³*DIIES university Mediterranea of Reggio Calabria 89122 Reggio Calabria, Italy*

⁴*Department of Psychology University of Milano Bicocca 20126, Milano, Italy*

⁵*ICAR Italian National Research Council (CNR) 87036 Rende (CS), Italy*

Abstract

Smart Workshops are experiencing the need of a mobile intelligence for mining both learning patterns and knowledge from the wide sea of data generated by both mobile users and mobile technologies. Indeed, mobile intelligence would represent the ideal substratum for providing "agentified" robots with a plethora of advanced capabilities (e.g., visual recognition, fault detection, self-recovery) and, hence, with high-level functionalities, like production line control, asset movement, connectivity restore. Besides the operational plane, however, mobile intelligence can be successfully exploited also in organizational tasks, like the formation of temporary, ad-hoc teams for accomplishing a given target. The complexity of some industrial operations, indeed, often demands the involvement of several, heterogeneous group of robots and the adequate representation of the reciprocal trustworthiness represents a key pre-requisite. It holds particularly for the Automated Guided Vehicles (AGVs) which are increasingly involved in collaborative activities aimed to optimise storage, picking, and transport functions in a wide variety of workshop areas. Therefore, in this paper we define a trustworthiness model for agentified AGVs based on the mix of their reputation and reliability and we present an agent-based framework implementing the related team formation strategy. The improvements obtained in terms of effectiveness and efficiency from the AGV team are observed and measured through a simulation activity, in which realistic settings for an industrial applications have been considered.

Keywords

Trust, Smart Factories, Team Formation, Multi-agent System.

1. Introduction

AGVs, namely fully autonomous robots able to operate without manual intervention or permanent conveying systems, are increasingly present in Smart Workshops. Just to name a few motivational examples, AGVs are ideal for replacing workers in repetitive, unappealing jobs as well as they push both speed and accuracy in moving products from shelf to shipping over human limits. Broadly speaking, AGVs lessen labour requirements and promise improving

WOA 2021: Workshop "From Objects to Agents", September 1–3, 2021, Bologna, Italy

✉ g.fortino@unical.it (G. Fortino); lidia.fotia@dimes.unical.it (L. Fotia); messina@dmi.unict.it (F. Messina); domenico.rosaci@unirc.it (D. Rosaci); giuseppe.sarne@unimib.it (G. M.L. Sarnè); claudio.savaglio@icar.cnr.it (C. Savaglio)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

effectiveness, efficiency and safety within the workshop area. In such a scenario, typical AGV applications include routine operations like the horizontal transport, storage and retrieval of materials as well as danger activities like clamp handling or extreme environmental conditions. In particular, AGVs result a critical enabling technology for agile production systems if devoted to collaborative tasks, such as the internal logistics ones. Therefore, the formation of temporary teams of heterogeneous AGVs is widely seen as an important advancement within the Industrial Internet of Things (IIoT) domain [1, 2, 3, 4]. However, establishing the criteria to rule such team formation process is challenging because of the mobility the AGVs, their different features (in terms of skills, autonomy, performances) and the potential lack of historical data or central shared repository. Therefore, more than exploiting structural or semantic similarities among team's partners, one can consider social properties existing among them for maximizing the probability of establishing positive interactions. In particular, a promising criterion consists in forming teams on the basis of the members' trustworthiness levels, namely the reliability shown in performing their own tasks and the reputation gained within the workshop area. Such two information, respectively, expressed in terms of efficiency and effectiveness, are usually embedded in a single measure named trust and can be shared within the workshop area (thus obviating the need of a centralized repository and also providing a higher fault tolerance, concurrency, etc.). To this end, a suitable solution is "agentifying" each AGV, leveraging on the widely established social, smart and cooperative attitudes of multi-agent systems (MAS) [5]. In particular, the agentified AGV can automatically update its trust information and the MAS can implement a team formation strategy by ranking AGVs based on their time availability (i.e., the time they need to accept a new task) suitably weighted by the trustworthiness value which, in its turn, embeds efficiency and effectiveness information combined accordingly to the factory policies. On these basis, we present our framework, more comprehensively described in [2], that leverages on a distributed MAS (to bypass the need of a central management system and its associated overhead, typically unacceptable for most industrial tasks) and on a trust model based on the mix of AGVs's reputation and reliability. We tested our team formation strategy on a simulated agent-based scenario, showing that combining mobile intelligence, team formation, reliability, reputation, and trust information leads to a measurable improvement of the simulated workshop area in terms of high quality performance.

The outline of the structured is as follows. An overview about agent-based technology, AGV and trust is reported in Section 2. The proposed framework is presented in Section 3 while Section 4 introduces the outlined trust model. Section 4.1 makes a connection between the contributions of these two Sections (i.e., the main architecture presented in Section 3 and the trust model of Section 4), showing how the trust model can be exploited to perform a team formation activity on the AGVs of the smart workshop in a distributed way. The results of our experiments are discussed in Section 5. Finally, in Section 6 conclusions are drawn.

2. Background and Related work

Multi-AGV systems have been recently adopted in the IIoT domain [1] to perform key activities like real-time monitoring, connectivity restore and collaborative control. Authors of [6, 7, 8], for example, illustrate some benefits provided by AGVs, in terms of reliability, efficiency and

safety, for the whole smart workshop area. If the advantages coming from the exploitation of teams of AGV are well-established, the discussion about the best criterion leading to the team formation process is still open. Indeed, more than conventional approaches based on geographical locality/social closeness or similarity (in terms of goals, skills, etc.), trustworthiness represents a novel, viable approach for dynamically and effectively grouping AGVs. Likewise, the agent-based computing (ABC) [5] is an enabling paradigm for information processing in dynamic, decentralized and scalable environments, where the entities exchange data to be automatically combined for outlining the global better setting (for example, resource allocation or scheduling problems). In particular, works like [9, 10] attest how trust systems have been widely implemented in the past through the ABC paradigm. Focused on the IIoT, our proposal relies on these research steps with the exploitation of AGVs which are enhanced through the multi-agent technology and are fully integrated within the trust system.

Along such research direction, a number of related work exist at the state-of-the-art[11, 12]. For example, authors [13, 14] sponsor the formation of group of both autonomously and cooperatively agent-based smart industrial devices for accomplishing tasks like controlling the materials handling and factory scheduling to automate the factory environment and its activities. In [15], instead, an agent-based controller is deputed to find the optimal, collision- and deadlock-free motion planning of its associated AGV. In the context of the Supply Chain Management, a real case study [16] showcases a framework integrating neutrosophic Decision Making Trial and Evaluation Laboratory technique with an analytic hierarchy process to effectively deal with uncertain and incomplete information. A recent work [17], instead, illustrates a combined solution exploiting simultaneously an OLE (Object Linking and Embedding) for process control technology, a software defined industrial network, and a device-to-device communication technology to achieve efficient dynamic resource interaction and management (to this end, an ontology modeling with multi-agent technology is used). The trustworthiness of potential partners is estimated in [18, 19, 20, 21, 22] through reputation systems based on first and second-hand information/observations, while in [23, 24, 25, 26, 27] by mainly analyzing the evolution of social relationships over time. In BETaaS [28], instead, a more comprehensive approach is presented, with a complex trust model for Machine-to-Machine applications taking into account factors like security [29, 30, 31, 32], QoS, scalability, availability and gateways reputation. In [33, 34], finally, cloud-based solutions to form groups of agentified industrial devices on the basis of their reliability and reputation values are presented.

With respect to these contributions, in our proposal the effective team formation is performed by means of a trustworthiness measure whose implementation (and the preliminary information exchange it requires) is enabled by the exploitation of the multi-agent technology in the entire framework, as detailed in the next Section 3.

3. Our Scenario

We consider a Smart Workshop adopting a swarm assembly approach with teams of coworkers for reaching the desired production goals in the required time. The considered scenario is modeled as follows:

- workshop's activities are performed by both human and AGVs, present in variable num-

- bers depending on the adopted processing technique;
- for each activity (e.g., welding, transportation, connectivity restore), there exists a specific kind of AGVs;
- AGVs differ with each other for efficiency (depending on their model, age, sensing capabilities, usury, etc.) and effectiveness (e.g., skills and so on) values;
- the agent of each AGV supports its physical counterpart for the working activity within the team of coworkers;
- a special *Manufacturing-Manager* (MM) agent is in charge of managing the production-lines and, in particular, of updating the measures of performance of the workshop agents and accordingly forming the “best” team/teams of AGV coworkers based on their trust measures;
- the MAS allows distributing the information load over the entire set of AGVs, thus avoiding the need of a unique, centralized repository.

In details, let W be the workshop area of our smart workshop and let SC be the daily set of customers requiring to the smart workshop the assembly of a customized item. Each customer $c \in SC$ has a reference to a MM agent, aiming at building for each item the best team/teams of AGVs capable of optimizing the production process in terms of performance. The MM periodically updates those two measures for each agent and, consequently, computes and updates the trustworthiness measure, (see Section 4). The MM saves a copy of these values in its internal memory, while each agent that has interacted with the MM saves a local copy of its measures. Therefore, when the agent will interact in the future with a novel MM, it will transmit the information about its efficiency, effectiveness and trustworthiness, as a sort of references.

4. The Trust Model

In this section, we introduce the trust model used to consider the performance of AGVs in a smart workshop. In this context, we define the following measures: the AVG *effectiveness* (γ) represents the customer satisfaction for the AVG’s job; in other words, it is the reputation that an AGV has in the customer community; the AGV *efficiency* (λ) represents the capability of complying with the product assembly constraints (e.g., time); in other words, it is the reliability with respect to the production-line operation; the AGV *trustworthiness* (τ) is a single trust measure that considers performance to properly guide the AGV team formation processes. τ combines *Efficiency* and *Effectiveness* to achieve a unique synthetic measure for a specific AGV.

In a controlled smart workshop, we assume that there are no malicious agents therefore it is not necessary to implement countermeasures against unsuitable behaviors (e.g., collusive, complainer, alternate, whitewashing and so on).

$\gamma \in [0, 1] \subset \mathbb{R}$ considers the feedback f_{eed} , with $f_{eed} \in [0, 1] \subset \mathbb{R}$, released by the customers to the AGV. More formally, γ is computed as:

$$\gamma^{new} = \beta \cdot \gamma^{old} + (1 - \beta) \cdot f_{eed} \quad (1)$$

where $\beta \in [0, 1] \subset \mathbb{R}$ is used to award a certain relevance to f_{eed} in updating γ with respect to its current value.

$\lambda \in [0, 1] \subset \mathbb{R}$ is calculated on the basis of objective measures (k) (e.g., the time required to complete a task) that can be combined in a single measure $\rho \in [0, 1] \subset \mathbb{R}$, with $\rho = f(k_1, \dots, k_n)$. More formally, λ is computed as:

$$\lambda^{new} = \alpha \cdot \lambda^{old} + (1 - \alpha) \cdot \rho \quad (2)$$

where $\alpha \in [0, 1] \subset \mathbb{R}$ is a parameter giving more or less relevance to ρ in updating λ with respect to its current value.

This trust model is the linear combination of reliability measures used with considerable results in our previous papers [33, 35] but contextualized in other multi-agent domains. Our proposal is adequate given the supposition that, if an increment of efficiency $\Delta\lambda$ (resp. effectiveness $\Delta\gamma$) produces an increase of trustworthiness $\Delta\tau$, then the percentage ratio $\frac{\Delta\tau}{\Delta\lambda}$ (resp. $\frac{\Delta\tau}{\Delta\gamma}$) should be the same for any increment of $\Delta\lambda$ (resp. $\Delta\gamma$). In Section 5, the experiments show that the linear model correctly reproduces the simulated scenario. More formally, τ is computed as:

$$\tau = \eta \cdot \lambda + (1 - \eta) \cdot \gamma \quad (3)$$

where $\eta \in [0, 1] \subset \mathbb{R}$ gives more or less relevance to λ with respect to γ ; η is set considering the factory policies in terms of performance. In our experiments (see Section 5), we have utilized a value $\eta = 0.4$ to give more importance to the effectiveness with respect to the efficiency.

4.1. Team formation

We recall that our trust model allows the team formation considering both present and past AVG results, in terms of performance. Each MM agent categorizes AGVs on the basis of the time need to accept a new task, called time availability TA , weighted on the τ value which embeds performance information combined accordingly to the factory policies. Therefore, AGV teams are formed by each MM selecting the top classified in this ranking. The set $G = \{g_0, g_1, \dots, g_n\}$ executes a distributed algorithm, where n is the total AVG numbers. In particular, g_0 is the MM agent and g_i is the i -th AGV agent. The algorithm is composed of five steps, called the *formation assignment*, the *request*, the *response*, the *selection*, and the *team formation*. The response step is executed by each agent g_i , $i = 1, \dots, n$, instead the MM agent g_0 performs the formation assignment, request, selection and team formation steps. In detail, the five steps operate as follows:

1. **formation assignment:** g_0 receives by its administrator (i.e., a human manager or a workflow process) the assignment to form a team. Then, g_0 produces as inputs for the step:
 - the agent's number z needful for the team formation;
 - the maximum waiting time t_{max} before starting the team formation;
 - the minimum trustworthiness τ_{min} required to an AVG for joining the team.
2. **request:** g_0 forwards a request to each agent g_i , $i = 1, \dots, n$ to obtain its time availability TA_i , representing the time that g_i needs to accept the step, and its trustworthiness τ_i .
3. **response:** an agent g_i computes the required values before providing a reply:

- TA_i based on the previous experiences (i.e., other steps in which it is previously involved);
- τ_i by combining efficiency and effectiveness (see Section 4).

Recall that each agent a_i continually updates the two measures according to both the time utilized to finish their steps and the feedbacks received by the customers. Then, TA_i and τ_i are sent to g_0 .

4. **selection:** g_0 continuously monitors the list R of the responses received by the AGV agents, containing the pairs (λ_i, γ_i) ; for each $i = 1, \dots, n$, g_0 calculates the following score:

$$R_i = \frac{1}{TA_i} \cdot \tau_i \quad (4)$$

Hence, g_0 deletes from R all those agents g_i whose $TA_i > t_{max}$ or $\tau_i < \tau_{min}$ because their AGVs are not eligible to perform the team formation. Also, g_0 stores R ordered by a decreasing value of the score R_i .

5. **team formation:** when t_{max} is reached, g_0 examines R and releases the following response to its administrator:
 - the list of the first z agents of R , if the cardinality of R is greater than or equal to z .
 - a failure message, otherwise.

All these steps are independently performed by the agents of the set G without the need of a central repository of the trustworthiness information regarding the AGVs. This choice allows to increase the efficiency of our model because the central repository management would imply a continuous updating of the AGV information with a consequent overhead for the internal communication network.

5. Experiments

The proposed industrial scenario has been simulated by a multi-agent system supporting the cooperation of AVG in a production site. To this aim, a significant number of workdays has been simulated by assuming that a random number of customer orders must be processed on each of the simulated workdays. Moreover, we assumed that the manufacturing process is organized in a serial way by production islands and in each production island one or more customization of the products are carried out according to the customers' order. On each island, the production process is carried out by a team of three smart AGVsc, denoted by heterogeneous performance, capable of autonomously operating.

In more detail:

- the heterogeneity of AGVs implies different skills and performance and, therefore, they will differ from each other also in terms of time required to complete the task assigned to them;
- given the different capabilities, AGVs will receive individual appreciation (i.e., the feedback f_{eed}) from the customer who placed the order for the customization work done.

As already described in Section 3, each production line is associated with an MM agent, who supervises the assembly of the items ordered by customers. In particular, each MM will interact with the software agents associated with AGVs to arrange the best AGV team for each specific production island (e.g., manufacturing task) at a given time with respect to both each specific order of a customer $c \in CS$ and the availability trustworthiness of each AGV. At the end of each production task, the parameters of effectiveness, efficiency, and trustworthiness are updated for each AGV.

The simulation has regarded a single smart production line for which the following parameters have been adopted: *i*) 60 working days; *ii*) 8 working hours for workday; *iii*) 150 customers' orders per workday; *iv*) 25 production islands¹ for each production line; *v*) 4 serial customized manufacturing tasks for each item and for each island; *vi*) 400 AGVs are active on the production-line, in other words 100 AGV for each of the 4 required manufacturing task to realize for item and for island. The parameters introduced above will drive both the operation of the production line and the response of the AGVs. In order to realize, to the best of our possibilities, a simulation as realistic as possible we have configured our production line adopting the most common parameters in use in some European factories that assemble cars.

Some preliminary tests have been carried out to suitably set the trust framework and, as a result of these tests:

- the parameters λ and γ were both initially set to 1.0 in order to assign maximum trustworthiness when reliability information is not yet available for then updating the AVG reliability based on subsequent experience;
- the parameter τ was initially set to 1.0;
- the parameters α and β were initially set to 0.95 (remember that λ and γ are updated through the feedback received over time in order to take into account even small variations in terms of performance)
- the parameter η (exploited to update the τ) were set to 0.4 conformly the criteria presented in Section 4.

Different scenarios were simulated by varying the performance of AVGs uniformly within suitable ranges of domains with the goal of forming efficient and effective AGV teams based on trustworthiness criteria. For this purpose, we considered the most critical scenario in our set of simulations, which is given by a combination of maximum performance loss varying from 5% to 25%. The results of these experiments are depicted in Figures 1 and 2.

Figure 1 depicts the changes in the parameters λ , γ and τ for the considered simulation period. It is evident how the proposed framework is able to produce significant advantages in terms of the plotted parameters. More specifically, note that the benefits in terms of λ (i.e., γ and τ) were evaluated incrementally based on the sum of the differences in efficiency (i.e., effectiveness and trustworthiness) measures of the AGV teams formed by applying the strategy proposed in Section 4.1 versus those that would have been formed based on temporal availability alone. Therefore, the results of this experiment show that our proposed trust framework allows improving both efficiency and effectiveness of the production line.

¹Remember that each production island is devoted to realize on or more (customization) tasks and an item will leave its current production island only after each manufacturing task of that island will end.

In contrast to the benefits described above, one must keep in account that the adoption of the proposed trust framework also has a cost in terms of average daily loss of time (in seconds) for AGV, which is depicted in Figure 2. This is due to the fact that the proposed strategy for forming teams is optimized with respect to the AGV's trust score (i.e., it takes into account AGVs' performance and time availability) and not on the basis of the only AGVs' time availability. This means that not the AGV with the best time availability is selected, but the one with the best placement resulting from a weighted average between performance and time availability (see section 4). More simulations have been performed to evaluate this "loss" of time, arriving to simulate also a periods up to 365 working days, achieving values that are always around the minute, on average compared to all the AGVs. This average time loss can be considered negligible in light of the improvement achieved in performance.

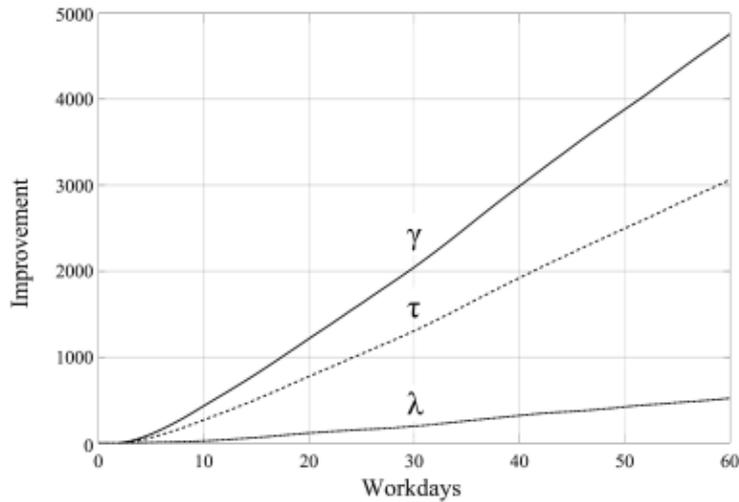


Figure 1: Advantage given by the trust framework in terms of λ , γ and τ .

6. Conclusions

The inherent complexity of many industrial activities demands for the cooperation of multiple, heterogeneous robots. In particular, teams of "agentified" AGVs with different capabilities are suitable candidates to accomplish both routine and extra-ordinary tasks by, simultaneously, improving the performance within a workshop area. Effectiveness and efficiency are two enabling factors for establishing trust among AGVs: therefore, in this paper, we have presented and tested a trustworthiness model and an agent-based framework to support the automatic formation of virtual, temporary teams of highly performing, mobile intelligent devices. The preliminary results obtained on a simulated industrial scenario with realistic settings have shown a measurable improvement in the teams composition in terms of both performance and appreciation. The implementation of the outlined agent-based framework, a parametric study of the trustworthiness model to achieve its best configuration and the introduction of management

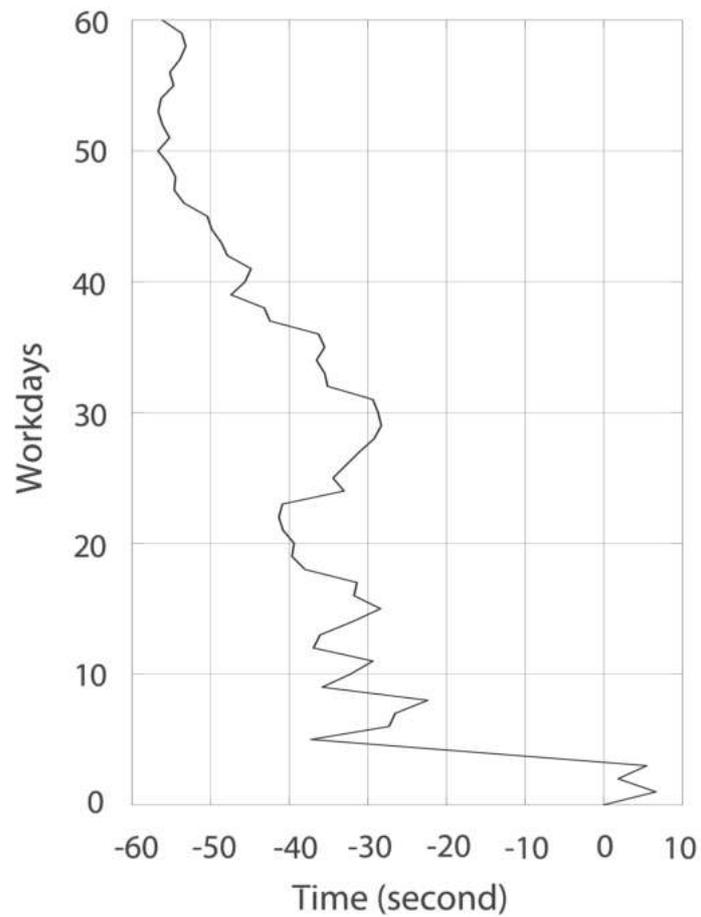


Figure 2: Average time “loss” for AGV.

techniques for handling unpredictable events potentially affecting the team formation represent our future research directions.

References

- [1] G. Fortino, C. Savaglio, M. Zhou, Toward opportunistic services for the industrial internet of things, in: 13th IEEE Conf. on Automation Science and Engineering (CASE), IEEE, 2017, pp. 825–830.
- [2] G. Fortino, L. Fotia, F. Messina, D. Rosaci, G. M. L. Sarné, Trust and reputation in the internet of things: state-of-the-art and research challenges, *IEEE Access* 8 (2020) 60117–60125.
- [3] J. Wan, J. Li, Q. Hua, A. Celesti, Z. Wang, Intelligent equipment design assisted by cognitive internet of things and industrial big data, *Neural Computing and Applications* 32 (2020) 4463–4472.

- [4] D. Uckelmann, M. Harrison, F. Michahelles, An architectural approach towards the future internet of things, in: *Architecting the internet of things*, Springer, 2011, pp. 1–24.
- [5] C. Savaglio, M. Ganzha, M. Paprzycki, C. Bădică, M. Ivanović, G. Fortino, Agent-based internet of things: State-of-the-art and research challenges, *Future Generation Computer Systems* 102 (2020) 1038 – 1053.
- [6] J. Theunissen, H. Xu, R. Y. Zhong, X. Xu, Smart agv system for manufacturing shopfloor in the context of industry 4.0, in: *Int. Conf. on Mechatronics and Machine Vision in Practice*, IEEE, 2018, pp. 1–6.
- [7] Y. Duan, Y. Luo, W. Li, P. Pace, G. Aloï, G. Fortino, A collaborative task-oriented scheduling driven routing approach for industrial iot based on mobile devices, *Ad Hoc Networks* 81 (2018) 86–99.
- [8] Y. Shi, X. Sun, W. Chen, R. Xie, X. Zheng, A co-evolutionary framework for concurrent design of machines layout and agvs planning, in: *2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, IEEE, 2017, pp. 498–502.
- [9] T. Huynh, N. Jennings, N. Shadbolt, An integrated trust and reputation model for open multi-agent systems, *Autonomous Agents and Multi-Agent Systems* 13 (2006) 119–154.
- [10] I. Pinyol, J. Sabater-Mir, Computational trust and reputation models for open multi-agent systems: a review, *Artificial Intelligence Review* 40 (2013) 1–25.
- [11] H. Yu, Z. Shen, C. Miao, C. Leung, D. Niyato, A survey of trust and reputation management systems in wireless communications, *Proceedings of the IEEE* 98 (2010) 1755–1772.
- [12] S. Ruohomaa, L. Kutvonen, E. Koutrouli, Reputation management survey, in: *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, IEEE, 2007, pp. 103–111.
- [13] C. Prasse, A. Nettstraeter, M. Ten Hompel, How iot will change the design and operation of logistics systems, in: *2014 International Conference on the Internet of Things (IOT)*, IEEE, 2014, pp. 55–60.
- [14] P. Farahvash, T. O. Boucher, A multi-agent architecture for control of agv systems, *Robotics and computer-Integrated manufacturing* 20 (2004) 473–483.
- [15] S. C. Srivastava, A. K. Choudhary, S. Kumar, M. Tiwari, Development of an intelligent agent-based agv controller for a flexible manufacturing system, *The International Journal of Advanced Manufacturing Technology* 36 (2008) 780.
- [16] M. Abdel-Basset, G. Manogaran, M. Mohamed, Internet of things (iot) and its impact on supply chain: A framework for building smart, secure and efficient systems, *FGCS* 86 (2018) 614–628.
- [17] J. Wan, B. Chen, M. Imran, F. Tao, D. Li, C. Liu, S. Ahmad, Toward dynamic resources management for iot-based manufacturing, *IEEE Communications Magazine* 56 (2018) 52–59.
- [18] A. Comi, L. Fotia, F. Messina, D. Rosaci, G. M. L. Sarné, Grouptrust: Finding trust-based group structures in social communities, in: *International Symposium on Intelligent and Distributed Computing*, Springer, 2016, pp. 143–152.
- [19] Y. Saied, A. Olivereau, D. Zeghlache, M. Laurent, Trust management system design for the internet of things: A context-aware and multi-service approach, *Computers & Security* 39 (2013) 15.
- [20] F. Messina, G. Pappalardo, A. Comi, L. Fotia, D. Rosaci, G. M. L. Sarné, Combining

- reputation and qos measures to improve cloud service composition, *International Journal of Grid and Utility Computing* 8 (2017) 142–151.
- [21] G. Fortino, L. Fotia, F. Messina, D. Rosaci, G. M. Sarné, A reputation mechanism to support cooperation of iot devices., in: *AI&IoT@ AI* IA*, 2019, pp. 28–39.
 - [22] G. Fortino, L. Fotia, F. Messina, D. Rosaci, G. M. Sarné, Using trust and “utility” for group formation in the cloud of things, in: *International Conference on Internet and Distributed Computing Systems*, Springer, 2019, pp. 112–122.
 - [23] F. Bao, I.-R. Chen, Dynamic trust management for internet of things applications, in: *Proc. of the 2012 int. work. on Self-aware internet of things*, ACM, 2012, pp. 1–6.
 - [24] P. De Meo, F. Messina, M. N. Postorino, D. Rosaci, G. M. L. Sarné, A reputation framework to share resources into iot-based environments, in: *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*, IEEE, 2017, pp. 513–518.
 - [25] E. M. Schooler, D. Zage, J. Sedayao, H. Moustafa, A. Brown, M. A., An architectural vision for a data-centric iot: Rethinking things, trust and clouds, in: *Distributed Computing Systems, 2017 IEEE 37th Int. Conf. on*, IEEE, 2017, pp. 1717–1728.
 - [26] R. Chen, F. Bao, J. Guo, Trust-based service management for social internet of things systems, *IEEE transactions on dependable and secure computing* 13 (2016) 684–696.
 - [27] L. Fotia, F. Messina, D. Rosaci, G. M. Sarné, Using local trust for forming cohesive social structures in virtual communities, *The Computer Journal* 60 (2017) 1717–1727.
 - [28] C. Vallati, E. Mingozzi, G. Tanganelli, N. Buonaccorsi, N. Valdambrini, N. Zonidis, B. Martínez, A. Mamelli, D. Sommacampagna, B. Anggorojati, et al., Betaas: A platform for development and execution of machine-to-machine applications in the internet of things, *Wireless Personal Communications* 87 (2016) 1071–1091.
 - [29] F. Buccafurri, L. Fotia, G. Lax, Social signature: Signing by tweeting, in: *International Conference on Electronic Government and the Information Systems Perspective*, Springer, 2014, pp. 1–14.
 - [30] F. Buccafurri, L. Fotia, G. Lax, Allowing continuous evaluation of citizen opinions through social networks, in: *International Conference on Electronic Government and the Information Systems Perspective*, Springer, 2012, pp. 242–253.
 - [31] F. Buccafurri, L. Fotia, G. Lax, Allowing non-identifying information disclosure in citizen opinion evaluation, in: *International Conference on Electronic Government and the Information Systems Perspective*, Springer, 2013, pp. 241–254.
 - [32] F. Buccafurri, L. Fotia, G. Lax, Allowing privacy-preserving analysis of social network likes, in: *2013 Eleventh Annual Conference on Privacy, Security and Trust*, IEEE, 2013, pp. 36–43.
 - [33] G. Fortino, F. Messina, D. Rosaci, G. M. L. Sarné, Using trust and local reputation for group formation in the cloud of things, *Future Generation Computer Systems* 89 (2018) 804–815.
 - [34] M. Chen, Y. Hao, C.-F. Lai, D. Wu, Y. Li, K. Hwang, redopportunistic task scheduling over co-located clouds in mobile environment, *IEEE Transactions on Services Computing* 11 (2016) 549–561.
 - [35] F. Messina, G. Pappalardo, C. Santoro, D. Rosaci, G. M. L. Sarné, A multi-agent protocol for service level agreement negotiation in cloud federations, *International Journal of Grid and Utility Computing* 7 (2016) 101–112.

Predicting humans: a sensor-based architecture for real time Intent Recognition using Problog

Gennaro Daniele Acciario¹, Fabio Aurelio D'Asaro² and Silvia Rossi¹

¹DIETI, University of Naples Federico II, Italy

²Logic Group, Department of Philosophy, University of Milan, Italy

Abstract

In a world where the population is aging, products that improve living comfort will have more importance in people's lives. These products must interpret the intentions of those who live in the house to provide them with assistance in their daily tasks. Motivated by these issues, we present an architecture for real-time *Intention Recognition*. We demonstrate it with a kitchen use-case, where the agent prepares a meal. Our goal is to recognize what type of meal the agent intends to prepare. The architecture consists of two layers, namely the "Classification Layer" and the "Problog Layer". The Classification Layer recognizes the environment through sensors and classifiers, and passes the information to the Problog Layer, which uses Problog to infer the intention. The Problog Layer consists of two Knowledge Bases: the "Static KB" and the "Dynamic KB". The former axiomatically describes the intentions we want to recognize, while the latter is generated at runtime using information from the Classification Layer.

Keywords

Intention Recognition, Problog, Knowledge Representation, Smart Technologies

1. Introduction

The latest version of "World Population Ageing" - an annual report of the United Nations¹ - outlines two meaningful statistics: in 2020, people over 65 are 727 million and are expected to increase to 1.5 billion by 2050. This same report mentions that, in most developed countries, these people will manage to live without a caregiver's external support, mainly thanks to good welfare and healthcare system. For these reasons, we can assume that soon it will be necessary to understand these people's intentions in an automated way to provide them with better comfort in a home environment through smart-home products designed to help these people in their daily tasks.

This paper focuses precisely on this aspect, presenting a logic-based architecture for intention recognition, which we demonstrate through a proof of concept. The use-case is that of smart kitchen environment, where our automated system aims to recognize what the human intends to cook - which is a particular instance of an *Intention Recognition* problem. According to [1],

WOA 2021: 22nd Workshop From Objects to Agents, September 01–03, 2021, Bologna, Italy

✉ fabio.dasaro@unimi.it (F. A. D'Asaro); silrossi@unina.it (S. Rossi)

🌐 <http://www.fabiодasaro.com> (F. A. D'Asaro)

🆔 0000-0002-2958-3874 (F. A. D'Asaro); 0000-0002-3379-1756 (S. Rossi)

© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

¹https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/undes_a_pd-2020_world_population_ageing_highlights.pdf

Intention Recognition is the process of becoming aware of the intention of another agent and, more technically, inferring an agent's intention through its actions and their effects on the environment. Hence, an intention is inferred from a sequence of actions. In this paper, we detect an action as the combination of two parts, namely the object the human is working with (e.g., milk, orange, knife) and the human's pose, which we consider in order to disambiguate what the agent is currently doing with the object (e.g., cutting, taking or pouring). In our architecture, two different Machine Learning-based classifiers detect these two parts of an action. We then feed the classifiers' outputs to a Problog architecture.

Since we are mainly concerned with sequences of actions, we chose to use a popular temporal ontology known as the Event Calculus [2, 3] which allows for the definition of events occurring along an explicit timeline. Moreover, given our Machine Learning classifiers and actions' probabilistic nature, we found it natural to use a probabilistic extension of this language. Among the possible choices [4, 5, 6] we picked the Problog-based system Prob-EC [6] as this has been successfully applied to similar use cases such as Event Recognition. It is worth noting here that, unlike the Event Recognition task, in Intention Recognition one aims to detect what the agent intends to do in the near future (e.g., "prepare a salad"), rather than an activity that is currently being performed (e.g., "two people are meeting each other").

It is worth noting that although in this paper we present a specific use-case of our architecture, this proof of concept may serve as a blueprint for applications in very different domains. For instance, a conversational chat-bot may want to track user activity in the calendar over multiple days to infer long-term intentions, e.g. it may deduce that the user intends to lose weight from the fact s/he has been exercising a lot and s/he's been buying low-fat foods for the last two weeks. It may also e.g. be employed by shopping centers as an anti-theft system that processes CCTV footage in real time. Furthermore, it could be used to provide both long and short-term assistance to the elderly, e.g. by understanding their intention and providing adequate assistance to finalize them. These use-cases are all very different from each other. However, as we will show in the following sections, they share a common structure: they all make use of time-stamped multimodal data which must be processed in order to deduce some form of user intention. This is precisely the type of problem our architecture aims to tackle. Given the agnostic nature of the building blocks of our architecture, we claim that our work can be readily generalized from our simple proof of concept to more complex domains.

This paper is organized as follows. In section 2 we shortly review related work. In section 3 we provide an overview of the architecture. In section 4 we explain in detail the technologies used to create the proposed architecture. In section 5 we demonstrate the architecture in our specific kitchen use-case. In Section section 6 we present some tentative conclusions and hint at future work.

2. Related Work

The problem of Intention Recognition is a significant one in the field of Human-Computer Interaction. It applies to a wide variety of tasks, ranging from smart homes [7] to neurosciences [8]. Intent Recognition has become an increasingly important field of research in recent years, and several papers have been published proposing different techniques and technologies to

approach it.

As Charniak indicated in 1993 [9], the nature of this discipline must be probabilistic. Bayesian networks are often used when working with uncertainty. Nazerfard and Cook [10] use Bayesian Networks with a continuous normal distribution to predict when the next intended action will occur. Pereira and Han [11] propose the use of Casual Bayesian Networks with plan generation techniques to predict hidden actions and unobservable effects. To a similar aim, Muncaster and Ma [12] propose Dynamic Bayesian Networks. However, in the context of Intent Recognition, Bayesian Networks have two particular problems:

- They do not allow for an explicit representation of a timeline,
- It is difficult to track the sequence of actions, which is central to the very nature of intentions.

For these reasons, we preferred a Probabilistic Event Calculus approach over Bayesian Networks.

Vilain [13] proposes using the analysis of an acyclic Context-Free Grammar to interpret sequences of steps, using a deductive process. The use of Spatial-Temporal And-Or Graphs (ST-AOG) was proposed in [14] and [15]. The ST-AOGs define the sub-activities constituting the final intention. In this paper, we predict agent intentions in Problog through probabilistic rules that correspond to the sub-tasks of an intention.

As we describe in the remainder of this paper, this paper uses two fundamental technologies: Problog and convolutional neural networks (CNN). Problog [16] is a Probabilistic Logic Programming Language with a Prolog-like syntax. Clauses can be decorated with a probability $p \in [0, 1]$ according to the following syntax:

$$p :: \text{Head} :- \text{Body}.$$

The Problog Layer of our architecture implements a probabilistic variant of the Event Calculus known as Prob-EC [6]. It consists of two Knowledge Bases: a Static KB (SKB) and a Dynamic KB (DKB). The SKB contains the domain independent axioms of Prob-EC and general static information about actions and intentions. The DKB is updated at runtime by translating classifier data into probabilistic events whenever the secondary server receives an action. The Problog Layer computes the probability of observing the sequence of action given each possible intention by querying the SKB and the DKB, and eventually outputs the intention(s) maximizing the corresponding likelihood.

As a running example, throughout the paper we use a set $\mathcal{I} = \{\text{Breakfast}, \text{Pesto Pasta}, \text{Tomato Pasta}, \text{Fruit Salad}, \text{Fish}\}$ of possible intentions. The Problog Layer calculates the likelihood of these activities when a sequence of actions is performed (e.g. take milk, pour it, and take cookies), and then informs the Main Server about the intention that maximizes such likelihood (e.g., preparing breakfast), which in turn displays it on the screen. In the case of a tie, we display all activities with maximal likelihood. A Convolutional Neural Network (CNN) is a typology of neural network able to perform classifications based on the operation of convolution between matrices.

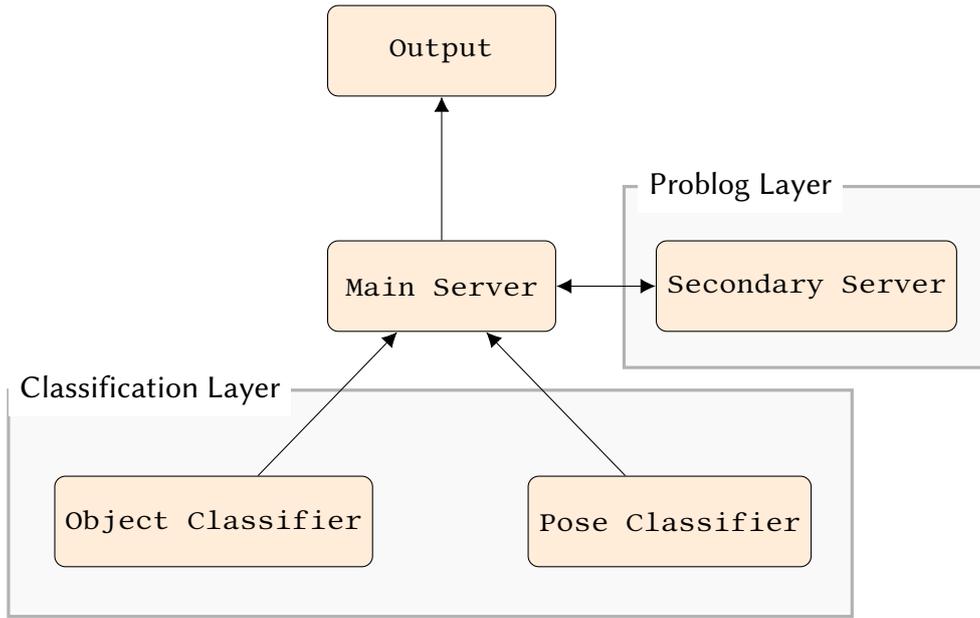


Figure 1: System Architecture

3. Architecture

Our simple proposed architecture for real-time intention recognition is shown in Figure 1. Our intention recognition system consists of two cameras. One camera, located near the human agent, is devoted to recognizing objects, with the other one installed further away from the human to recognize its pose. The Classification Layer of our architecture processes the video stream captured from the cameras. It extracts the object the human agent is currently using, and her pose. Then, it forwards this information to the main server. To prevent flooding, we set a minimum delay of 0.5 seconds between requests to the main server.

The main server stores the data in a buffer, and when the buffer is full, it selects the most frequent action in the buffer and sends it to a secondary server. When the secondary server receives an action, it translates the action into a Problog probabilistic fact and adds it to the (Dynamic) Knowledge Base. It then compiles the whole script and queries it to figure out the most probable intention. We implemented communication between layers through HTTP calls. In particular, the output of the secondary server is a JSON created from the output values that the script in Problog returns. In the remainder of this section, we discuss each component of our system in greater detail.

3.1. Physical Setup

Our controlled environment consists of a video camera (used by the object classifier) on the working table, facing the agent. The other video camera (used for pose recognition) is on the

working table, approximately at 1.8 meters from the agent.

3.2. Classification Layer

The Classification Layer consists of two classifiers that receive the video stream from the two cameras. One of the classifiers aims to recognize objects, with the other one recognizing the agent's pose.

The object classifier is based on MobileNet [17], a Convolutional Neural Network that uses a technique called "Depthwise Separable Convolution" to reduce the computational cost of convolution [17]. Two hyperparameters allow one to further improve MobileNet's computational efficiency, namely the Width Multiplier α and Resolution Multiplier ρ , that optimize the model according to the context.

We used PoseNet [18] to perform the pose recognition task. It supports recognition algorithms both for a single person and for several people simultaneously. PoseNet recognizes 17 key points corresponding to important points of the human skeleton. It associates spatial coordinates to each keypoint, which it then further processes to classify the user's pose. We chose these two models due to their simplicity in performing class training. Nonetheless, the architecture proposed in this paper is also scalable with respect to several other technologies or alternatives that perform the same purposes of object and pose recognition.

4. Implementation

As mentioned above, the Problog script consists of two knowledge bases: the SKB and DKB.

The DKB gets updated every time the classifiers detects an object or a gesture. For instance, if the pose classifier detects that the human agent is performing the gesture *take* at time *t*, we augment the DKB with the following probabilistic fact:

$$p :: \text{happensAt}(\text{gesture}(\text{take}), t)$$

where *happensAt* is a standard Prob-EC predicate to handle event occurrences, and *p* is the recognition probability associated to the gesture *take* by the pose classifier. Similarly, if the object classifier detects that the human agent is interacting with the ingredient *apple* at time *t*, we translate this to:

$$p :: \text{happensAt}(\text{ingredient}(\text{apple}), t)$$

On the other hand, the SKB defines how the probability of an intention increases as the result of recognizing an object and/or an action, as in the following example:

$$0.2 :: \text{initiatedAt}(\text{breakfast} = \text{true}, T) :- \text{take}(\text{milk}, T).$$

$$0.5 :: \text{initiatedAt}(\text{breakfast} = \text{true}, T) :- \text{takeAndPour}(\text{milk}, T).$$

where *initiatedAt* is a standard Prob-EC predicate to quantify how an event occurrence affects the probability of a *fluent*, i.e., a property of the world, which in our example is the intention

breakfast to be recognized. The two predicates *take* and *takeAndPour* are abbreviations defined as follows:

$$take(Obj, T) :- happensAt(gesture(take), T), happensAt(ingredient(Obj), T)$$

$$takeAndPour(Obj, T) :- take(Obj, Tprec), pour(Obj, T), Tprec < T$$

Finally, in order to query the likelihood of preparing *breakfast* at time t we use the in-built Problog predicate *query* as follows:

$$query(holdsAt(breakfast = true, t))$$

We query the SKB and the DKB in order to get the intention that maximizes the likelihood, and pass it on to the main server, which displays it on the screen. In the case of a tie, we display all activities with maximum likelihood.

5. Demonstration

In this section, we demonstrate how our architecture behaves in a few controlled experiments. We first set up the Problog SKB with reasonable probabilities associated with actions and intentions. Then, we let a human agent perform a series of actions. The system analyzed the video streams as outlined in section 3 and the Problog Layer produced the corresponding DKB. In each of the following subsections, we focus on specific experimental runs, by providing the DKB and showing how the probability of intentions evolves over time.

5.1. Equally likely intentions

Figure 2 shows one case in which the architecture is unable to disambiguate between two possible intentions up until time point 4. Note that in this example, the ingredients and their associated actions are sequences of actions that may constitute an intent. As you can see by observing "Tomato pasta" our probabilities are monotonous because we do not exclude the ambiguous case in which the user wants to return to an intention previously started and not concluded.

In this experiment, the sequence of events was as follows: the human agent took the water at times 0 and 1, poured it at time 2, and then took the pasta at time 3. This narrative is captured by the events generated by the Problog Layer, which in this case are as follows:

1 :: *happensAt(gesture(take), 0)*.

1 :: *happensAt(ingredient(water), 0)*.

1 :: *happensAt(gesture(take), 1)*.

1 :: *happensAt(ingredient(water), 1)*.

1 :: *happensAt(gesture(pour), 2)*.

1 :: *happensAt(ingredient(water), 2)*.

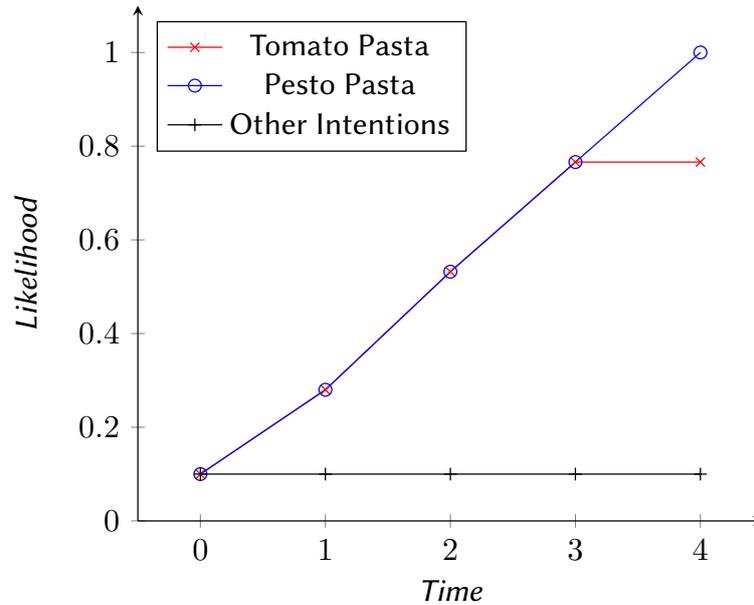


Figure 2: At time-points 0, 1, 2 and 4 the human agent performs actions that are compatible both with the intention of preparing *Tomato Pasta* and *Pesto Pasta*. However, at instant 4 the agent *takes* the ingredient *pesto*, making *Pesto Pasta* the most likely intention. All other intentions are considered to be very unlikely at all time points.

- 1 :: *happensAt(gesture(take), 3)*.
- 1 :: *happensAt(ingredient(pasta), 3)*.
- 1 :: *happensAt(gesture(take), 4)*.
- 1 :: *happensAt(ingredient(pesto), 4)*.

At time 3, the system is unable to figure which type of pasta the agent intends to prepare. This can be clearly seen from the figure, which shows the systems assigns equal likelihood to the intention of preparing *Tomato Pasta* and *Pesto Pasta*. However, as soon as the human agent took the pesto (time 4), the system was able to determine that her intention is that of preparing *Pesto pasta*.

5.2. Time factor

Figure 3 shows how Prob-EC allows us to overcome one of the problems affecting Bayesian Networks in an Intention Recognition setting, i.e. the management of the temporal factor. In this example, the human agent has an interaction with ingredient *milk* lasting 4 time points. This is encoded in the following DKB:

- 1 :: *happensAt(gesture(take), 0)*.
- 1 :: *happensAt(ingredient(milk), 0)*.

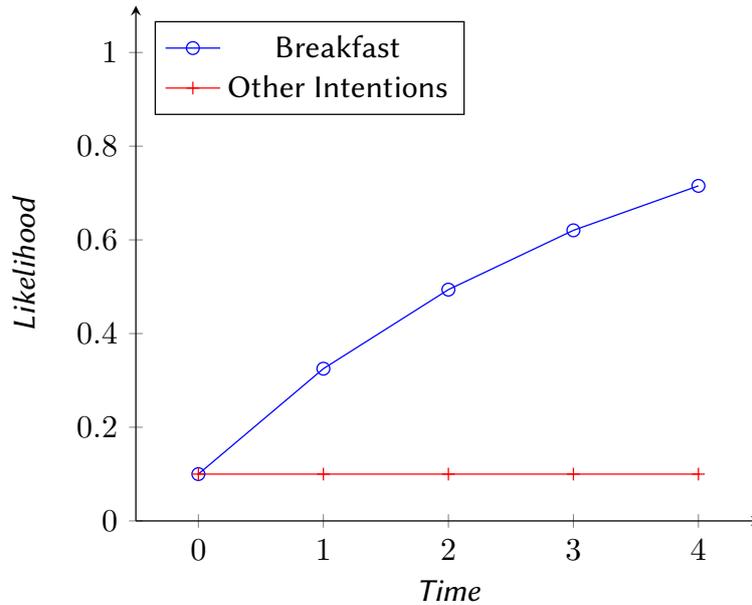


Figure 3: In this example, the agent interacts with the ingredient *milk* at instants 0, 1, 2 and 3. As the *Breakfast* intention becomes more likely, the other intentions remain unlikely as they are incompatible with the use of *milk*.

- 1 :: *happensAt(gesture(take), 1)*.
- 1 :: *happensAt(ingredient(milk), 1)*.
- 1 :: *happensAt(gesture(take), 2)*.
- 1 :: *happensAt(ingredient(milk), 2)*.
- 1 :: *happensAt(gesture(take), 3)*.
- 1 :: *happensAt(ingredient(milk), 3)*.

In this case it is reasonable that the longer the agent interacts with the milk lasts, the greater its intention to have breakfast. Our system behaves accordingly, as shown in fig. 3.

5.3. The complete use case

In previous examples, we had 100% recognition accuracy attached to all events. This was to show how our system behaves when classifiers do not have an associated classification accuracy. We now look at a case where the probability of facts may vary according to classification accuracy, as in the case of our specific system.

In the following experiment, we asked the human agent to perform actions as she normally would when preparing breakfast. She held the milk for two time points (with actions recognized with 85% and 96% accuracy, respectively). Due to a classification problem, the system recognized an orange (78% accuracy) at time 2. Then, she poured the milk (93% accuracy), and then temporarily abandoned his main intention to read the expiration date of a jar of pesto

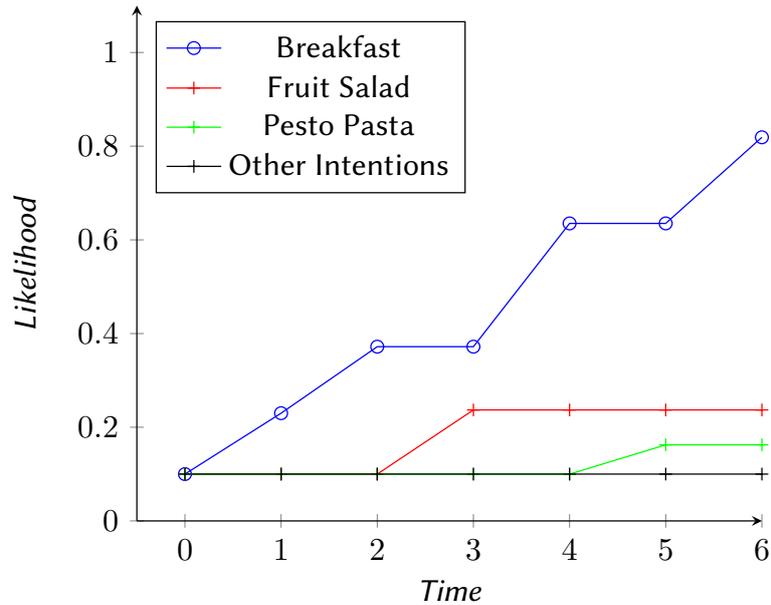


Figure 4: In this example, we show how the architecture behaves in a more realistic use case. The intention of preparing *Breakfast* is correctly recognized at all time points, in spite of a classifier error and the user temporarily performing another task.

(68% accuracy) before grabbing cookies (91% accuracy) to finalize the intention of preparing breakfast. The associated DKB was as follows:

- 0.84 :: *happensAt(gesture(take), 0)*.
- 0.86 :: *happensAt(ingredient(milk), 0)*.
- 0.92 :: *happensAt(gesture(take), 1)*.
- 0.98 :: *happensAt(ingredient(milk), 1)*.
- 0.78 :: *happensAt(gesture(take), 2)*.
- 0.81 :: *happensAt(ingredient(orange), 2)*.
- 0.93 :: *happensAt(gesture(pour), 3)*.
- 0.89 :: *happensAt(ingredient(milk), 3)*.
- 0.68 :: *happensAt(gesture(take), 4)*.
- 0.76 :: *happensAt(ingredient(pesto), 4)*.
- 0.91 :: *happensAt(gesture(take), 5)*.
- 0.94 :: *happensAt(ingredient(cookies), 5)*.

Figure 4 shows the results in this case. Note that the intention of preparing *Breakfast* is correctly recognized at all time points.

6. Conclusion and Future Work

This paper proposes the application of probabilistic logic-based architectures, more specifically Problog and Prob-EC in our case, to the task of Intention Recognition. As demonstrated in our example, we believe such tools may prove highly effective and impactful. Although similar approaches have been proposed for Event Recognition, using Event Calculus based architectures for real-time recognition of agents' intention may open up new possibilities and overcome some difficulties with other techniques. Our proposed architecture for a use-case of a smart kitchen can be seen in fig. 1. It includes two main layers: the Classification Layer, sensing the environment, and the Problog Layer, which performs logic-probabilistic inference to derive the most likely intention of the user. In this work, we provide a proof of concept that mainly shows how our architecture works in a series of controlled experiments. Nonetheless, this very architecture may be generalized to other use-cases. The next step of this research will involve human judgment to systematically evaluate the detection accuracy of intention. Furthermore, we foresee that such an architecture might suit the task of learning and predicting complex intentions that were not described a priori. In the future, we aim to further explore these applications and extensions. Finally, we aim to extend the use case to other objects and poses in order to be able to evaluate the performance of the system with respect to the classification of intentions.

References

- [1] T. A. Han, *Intention Recognition, Commitment and Their Roles in the Evolution of Co-operation From Artificial Intelligence Techniques to Evolutionary Game Theory Models*, Springer Berlin Heidelberg, 2013.
- [2] R. Kowalski, M. Sergot, A logic-based calculus of events, in: *Foundations of knowledge base management*, Springer, 1989, pp. 23–55.
- [3] R. Miller, M. Shanahan, Some alternative formulations of the event calculus, in: *Computational logic: logic programming and beyond*, Springer, 2002, pp. 452–490.
- [4] A. Skarlatidis, G. Paliouras, A. Artikis, G. A. Vouros, Probabilistic event calculus for event recognition, 2013. [arXiv:1207.3270](https://arxiv.org/abs/1207.3270).
- [5] F. A. D'Asaro, A. Bikakis, L. Dickens, R. Miller, Probabilistic reasoning about epistemic action narratives, *Artificial Intelligence* 287 (2020). URL: <https://www.sciencedirect.com/science/article/pii/S0004370219300906>. doi:<https://doi.org/10.1016/j.artint.2020.103352>.
- [6] A. Skarlatidis, A. Artikis, J. Filippou, G. Paliouras, A probabilistic logic programming event calculus, *Theory and Practice of Logic Programming* 15 (2015) 213–245. doi:[10.1017/S1471068413000690](https://doi.org/10.1017/S1471068413000690).
- [7] J. Rafferty, C. D. Nugent, J. Liu, L. Chen, From activity recognition to intention recognition for assisted living within smart homes, *IEEE Transactions on Human-Machine Systems* 47 (2017) 368–379. doi:[10.1109/THMS.2016.2641388](https://doi.org/10.1109/THMS.2016.2641388).
- [8] M. Iacoboni, I. Molnar-Szakacs, V. Gallese, G. Buccino, J. C. Mazziotta, G. Rizzolatti,

- Grasping the intentions of others with ones own mirror neuron system, *PLoS Biology* 3 (2005). doi:10.1371/journal.pbio.0030079.
- [9] E. Charniak, R. Goldman, A bayesian model of plan recognition, *Artificial Intelligence* 64 (1993) 53–79. doi:10.1016/0004-3702(93)90060-0.
- [10] E. Nazerfard, D. Cook, Using bayesian networks for daily activity prediction, *AAAI Workshop - Technical Report* (2013) 32–38.
- [11] L. Pereira, T. A. Han, Intention recognition via causal bayes networks plus plan generation, volume 5816, 2009, pp. 138–149. doi:10.1007/978-3-642-04686-5_12.
- [12] J. Muncaster, Y. Ma, Activity recognition using dynamic bayesian networks with automatic state selection (2007). doi:10.1109/WMVC.2007.5.
- [13] M. Vilain, Getting serious about parsing plans: A grammatical analysis of plan recognition, in: *Proceedings of the Eighth National Conference on Artificial Intelligence - Volume 1, AAAI'90*, AAAI Press, 1990, p. 190–197.
- [14] S. Holtzen, Y. Zhao, T. Gao, J. B. Tenenbaum, S.-C. Zhu, Inferring human intent from video by sampling hierarchical plans, *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016). doi:10.1109/iros.2016.7759242.
- [15] S. Qi, S. Huang, P. Wei, S.-C. Zhu, Predicting human activities using stochastic grammar, *2017 IEEE International Conference on Computer Vision (ICCV)* (2017). doi:10.1109/iccv.2017.132.
- [16] L. De Raedt, A. Kimmig, H. Toivonen, Problog: A probabilistic prolog and its application in link discovery., in: *IJCAI*, volume 7, Hyderabad, 2007, pp. 2462–2467.
- [17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. arXiv:1704.04861.
- [18] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, K. Murphy, Towards accurate multi-person pose estimation in the wild, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4903–4911.

Smart Balancing of E-scooter Sharing Systems via Deep Reinforcement Learning

Gianvito Losapio¹, Federico Minutoli¹, Viviana Mascardi¹ and Angelo Ferrando¹

¹DIBRIS, University of Genova, Italy

Abstract

Nowadays, micro-mobility sharing systems have become extremely popular. Such systems consist in fleets of electric vehicles which are deployed in cities, and used by citizens to move in a more ecological and flexible way. Unfortunately, one of the issues related to such technologies is its intrinsic load imbalance; since the users can pick up and drop off the electric vehicles where they prefer. We present ESB-DQN, a multi-agent system based on Deep Reinforcement Learning that offers suggestions to pick or return e-scooters in order to make the fleet usage and sharing as balanced as possible.

Keywords

Micro-mobility, E-scooter Sharing Systems, Multi-agent Systems, Deep Reinforcement Learning

1. Introduction

In the last few years, micro-mobility sharing systems have become extremely popular. More and more companies are purchasing fleets of electric vehicles to be deployed in many cities around the world, allowing users to easily rent vehicles via a smartphone app. The last trend is to offer a so-called "free-floating" or "dockless" service related to e-scooters, e-bikes or e-moped: the vehicles can be picked-up or dropped-off anywhere within an operative area designed by the service provider to cover most of the busiest areas of cities [1, 2].

The great flexibility of such a service comes with the challenge of unpredictable usage patterns, with the result of an imbalanced distribution of the electric vehicles around the city. Moreover, battery capacity is limited and many vehicles can rapidly become out-of-charge during the course of the day, if overused in quick succession. In order to preserve a good quality of service despite of imbalance problems and battery limitations, companies need to devote a large operational effort for an efficient fleet management [3].

Typically, specialised workers are employed to accomplish two different, yet complementary tasks, namely *battery swap* and *relocation*. *Battery swap* refers to the process of inserting new batteries into out-of-charge vehicles, whereas *relocation* refers to the process of moving vehicles from one zone to another in order to rebalance the fleet distribution [4].

Quantity of workers, modality and frequency associated to *battery swap* and *relocation* operations represent crucial aspects in the definition of an efficient fleet management policy. A

WOA 2021: 22nd Workshop "From Objects to Agents", Bologna, Italy, 1-3 September 2021

✉ gvlosapio@gmail.com (G. Losapio); fede97.minutoli@gmail.com (F. Minutoli); viviana.mascardi@unige.it (V. Mascardi); angelo.ferrando@unige.it (A. Ferrando)

ORCID  0000-0002-1024-7512 (G. Losapio); 0000-0002-5472-7673 (F. Minutoli)

 © 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

critical trade-off is required to avoid high operational costs and, at the same time, maximize the usage of vehicles. Recently, users engagement has been proposed as a viable solution to alleviate the aforementioned problems. As a result, nowadays several companies engage users in various ways to solve the imbalance and the battery limitation problems [5, 6, 7].

In this work, we present ESB-DQN, a multi-agent system based on Deep Reinforcement Learning (Deep RL) capable of proposing convenient alternative locations for picking up or returning e-scooters. Every time a user is willing to rent an e-scooter, he/she is encouraged to accept alternative pick-up or drop-off points in exchange for monetary incentives.

Based on demand forecast models and artificial intelligence techniques, the ESB-DQN system is able to learn convenient recommendations for the users, in order to maximize the vehicle availability and, at the same time, minimize the number of *battery swap* and *relocation* operations. As a result, the system is able to improve service efficiency and to increase the service provider's long-term revenue. Provided with a smart monetary incentive mechanism, the system is also intended to improve customers' satisfaction and fidelity.

The code that supports the findings of this study is available upon request.

The main contributions of our paper are the following:

- an innovative customer-oriented rebalancing strategy has been defined through a multi-agent system based on deep reinforcement learning;
- an existing simulator of mobility sharing systems has been integrated with a state-of-the-art library for deep reinforcement learning;
- simulations based on real data have been carried out to preliminarily quantify the benefits of the proposed approach.

The paper is organized as follows: Section 2 contains an overview of related works. Section 3 describes the materials and the methods used throughout this research. Section 4 presents the experiments that have been carried out as well as the corresponding results. Section 5 concludes the paper with discussions and possible future works.

2. Related works

The recent work by Wen and colleagues [7] provides a comprehensive overview of the rebalancing strategies used to alleviate the imbalance problem in bike sharing systems. Such strategies have been classified according to two main categories: truck-based rebalancing and customer-oriented rebalancing. Truck-based rebalancing refers to the *relocation* operations mentioned above in Section 1. A specialized group of workers is in charge of moving vehicles from one zone to another by means of trucks. On the other hand, customer-oriented rebalancing is the process of encouraging users to adopt efficient behaviours by providing incentives. The latter category is the main topic of our investigation.

Most past works on rebalancing strategies are not targeted towards "free floating" systems. In particular, papers investigating truck-based rebalancing determine the optimal inventory for each station and design a dynamic optimal truck route with budget constraint [8, 9]. Analogously,

paper investigating customer-oriented rebalancing ponder the role of stations in the incentive proposals mechanism [5, 6, 7].

In our work, both rebalancing strategies have been taken into account: truck-based rebalancing is implemented through the simulator, whereas customer-oriented rebalancing is implemented through the reinforcement learning system. Few other works employ deep RL to investigate user incentives in bike sharing systems, including [10, 11]. However, their objective is to determine an optimal pricing mechanism, whereas the objective of our work is to determine convenient pick-up/drop-off zones for each booking request.

The motivation behind the use of deep Reinforcement Learning for such a task is mainly related to the possibility of combining many interesting aspects at once. The deep RL system can indeed incorporate demand forecasting models as a baseline to drive agents' behaviours and, at the same time, can learn efficient suggestions based on past experience and adapt to real-time demand and availability of the system. In this way, the decision process behind the offered suggestions can capture complex information about the dynamics of the mobility system. Furthermore, by formulating the problem as a game, several constraints may be introduced to enforce specific objectives in the mobility system (e.g., a target service availability).

Compared to previous works, the innovative contribution of our paper is thus twofold. On one side, the imbalance problem inside "free-floating" e-scooter mobility systems has been addressed for the first time. Both rebalancing strategies proposed so far in the literature have been adapted from station-based sharing systems. On the other side, a deep RL multi-agent system in charge of suggesting pick-up/drop-off zones constitutes an original solution which does not build on any existing work. The main influential work has been [4], in which the simulator has been introduced (Section 3.2) - an essential component of the ESB-DQN system.

3. Materials and methods

3.1. Data

To investigate the free-floating imbalance problem, we rely on actual e-scooter trips open data published by the Municipality of Louisville¹, Kentucky. The data comes fuzzed both in time and space for privacy reasons; in particular, each trip has any time-related information rounded to the closest quarter of an hour and any space-related information rounded at the 3rd decimal for both latitude and longitude. Hence, we follow the disaggregation procedure described in [4] such that each trip retains a unique Id, the duration, the distance, the start time, the end time, the start location and the end location. The main characteristics of the dataset are summarized in Table 1. They refer to a *training window* of observations registered over the whole year 2019. The number of trips in the simulation, denoted as $N \text{ trips sim}$ refers to a simulation window of observations over one single day, namely January 01, 2020.

Louisville's e-scooter ecosystem has rather limited complexity, reflecting heterogeneous temporal and spatial demands at the same time. Nonetheless, in order to further ease the formulation of the problem, the whole operative area in the city of Louisville has been quantized in a set \mathcal{Z} of $l \times l$ square zones as proposed in [4], with l the side of the squares, a key parameter

¹<https://data.louisvilleky.gov/dataset/dockless-vehicles>

Table 1
Main characteristics of Louisville dataset

City	N scooters	Avg trip dur. (s)	Avg trip dist. (m)	N zones	N trips train	N trips sim
Louisville	800	814	1601	279	199 789	154

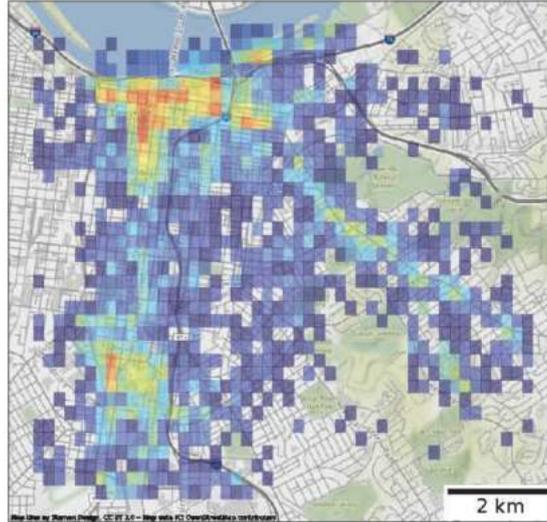


Figure 1: Characterisation of zones in the city of Louisville, following a heatmap colour scale on the number of average trip requests per zone in 2019. Credits to [4].

later clarified in Section 4. As a result, the start/end locations of each trip report the Id of the corresponding zone membership. Each zone $z_i \in \mathcal{Z}$ is associated with a set of valid 1-hop neighbours \mathcal{N}_{z_i} , i.e. the zones among the 8 adjacent zones registering at least one booking request within the *training window* of observations. As it can be seen in Figure 1, many of the zones do not have a full set of valid 1-hop neighbours, i.e. $|\mathcal{N}_{z_i}| \neq 8$. In fact, almost all of them do not, with a grand total space of valid neighbours, \mathcal{N}_{valid} , amounting to only the 60.6% of the whole space of possible neighbours \mathcal{N}^* , with $|\mathcal{N}^*| = 279 * 8 = 2232$.

Both the large subset of invalidity, $\mathcal{N}_{invalid}$, and the tiny simulation window of observations over January 01, 2020 only, are further discussed in Section 4 as they play a key role in the reasoning behind the training of the ESB-DQN multi-agent system.

3.2. Simulator

A modified version of the SimPy-based simulator presented in [4] has been used to simulate e-scooter sharing system dynamics in Louisville. A formal description of the simulator follows:

Fleet and zones. Let \mathcal{S} be the fleet of e-scooters. At any time t , each e-scooter $s \in \mathcal{S}$ is characterised by a unique plate Id, the state of availability, the state of charge of the battery

$b(s) \in [0, B]$, with B being the battery capacity, and the location $l(s)$ as a zone Id in \mathcal{Z} .

Trip requests. The simulator processes trip request events by directly reading them from the input trace over 2019. When the i -th trip request event fires at time t_i , the simulator checks whether there is any e-scooter s with enough residual energy, i.e., $b(s) \geq e_i$, being e_i the energy to complete such trip, either available in the same zone or in the 1-hop neighbouring zones (i.e., the 8 surrounding zones). This is equivalent to assume that customers will by default rent the nearest available e-scooter having enough battery charge.

Incentive proposals. In alternative, users are incentivized to pick-up and/or drop-off the vehicle from/to a different zone (in a limited nearby area). They randomly accept or decline the proposal according to a willingness factor $w \in [0, 1]$, and eventually get their incentive once the trip has been completed. If no alternative pick-up proposal is accepted and no scooter is available in the 1-hop neighbouring zones, the trip request is marked as *unsatisfied*.

Trip completion. Once the pick-up zone $p(i)$ and the drop-off zone $d(i)$ are defined, a trip-end event is scheduled at time $t_i + \delta t_i$, being δt_i the duration of the rental - drawn from a Gaussian distribution with mean μ equal to the duration of the trip reported in the trace, and standard deviation σ equal to 4 minutes, as a form of variability. When the trip-end event fires, the simulator makes the e-scooter s back available in position $d(i)$, and updates its battery charge $b(s) = b(s) - e(i)$. If $b(s) < \alpha B$, with α the operability threshold $\in [0, 1]$, the scooter s is marked as *dead* and is no longer available until a battery swap operation is performed.

Battery swap. Once every T_s time steps, a fleet of n_{swap} battery swap workers is triggered to perform battery swaps operations. Each worker is assigned a battery swap schedule, which consists of up to n_v vehicles to be re-charged inside several zones. Battery swap schedules are created and assigned with the following criteria: we compute the battery charge deficit for each zone z at time t , $\Delta_s(t, z)$, with $t = kT_s$, as the number of dead vehicles waiting for service in z . Then the zone z_o with the least deficit is identified, and a priority 0 queue is constructed for all the other $n - 1$ zones, with priority defined as:

$$p(t, z) = \frac{1}{\Delta_s(t, z)} + \frac{d(z, z_o)}{\max(d(z_j, z_o)_{j=1, \dots, n})}$$

with $d(z_i, z_k)$ being the Haversine distance between the i -th and the k -th zone. Each worker is then assigned a subset of the queue, potentially across multiple zones, following a lump sum costs policy whose goal is to construct a schedule that keeps the expected profit in the next T_s time steps, $P_{\text{swap}, t+T_s}$, higher than the expected battery swap costs, $C_{\text{swap}, t}$: an average cost of service has to be assumed for each vehicle, C_v . As soon as all the workers have an assigned schedule as a sequence of zone Ids, the shortest path to completion is computed for each of them by solving an equivalent TSP optimization problem. Once all the battery swap operations are completed, the workers wait as idle in their last zone on schedule.

Relocation. Once every T_r time steps in a limited working time interval T_{work} , a fleet of n_{rel} relocation workers is triggered to perform relocation operations. Each worker is assigned a

relocation schedule, which consists of up to n_v vehicles to be moved from some zones to others in order to balance the system. Relocation schedules are created following a similar criteria to what has been described above: a deficit $\Delta(t, z)$ is computed for each zone z , a priority 0 queue is computed off of that and a number of schedules are first generated following a lump sum costs policy and then optimized via TSP. In this case, $\Delta(t, z)$ is computed observing the availability of e-scooters with respect to the expected inward and outward flows for the zone z at time t computed over 2019 following the predictive model proposed in [4].

Initialization. At start time, e-scooters are randomly placed among the zones of the grid with uniform random charge $b(s) \in [B/2, B]$. Afterwards, both relocation and battery swap workers are similarly placed with uniform random among the 30 zones that have registered the highest demand in the training data over 2019. This is equivalent to assume the existence of landmarks within the city of Louisville that require a higher concentration of e-scooters.

Originally, battery swap operations were treated differently from relocation ones, as battery swap workers were modelled as a FIFO queue that would react on the fly to out of charge events. In this work, we have leaned towards the hourly scheduling approach already followed by relocation workers, as this would allow us to have a rough idea of the hourly workforce of battery swap workers that is necessary to do any sort of planning whose long-term objective is to reduce the overall maintenance costs of the system.

3.3. ESB-DQN multi-agent system

A multi-agent system has been designed, in charge of proposing alternative pick-up/drop-off zones to the users in change of incentives. In particular, two agents are defined: a *pick-up agent*, P , and a *drop-off agent*, D . At every generated trip request i with pick-up zone $p(i)$ and drop-off zone $d(i)$, the pick-up agent proposes an alternative pick-up zone $\hat{p}(i)$, whereas the drop-off agent proposes an alternative drop-off zone $\hat{d}(i)$. Both proposals share the same ultimate goal of improving the long-term balance of the system, while reducing the overall costs of service due to general maintenance, battery swap ops and relocation ops.

The next three paragraphs describe the fundamental components of the E-scooter Balancing DQN, or ESB-DQN for short, multi-agent system.

3.3.1. Environment

The environment wraps the modified simulator described in Section 3.2 to make it compliant with DeepMind’s DQN Zoo library for reinforcement learning [12]. The major change we have made to said simulator is conceptual: rather than simulating the whole cascade of trip requests between two time intervals of start and finish, t_0 and t_N , collecting a certain number of statistics about the run afterwards, as the original in [4] does, the simulator moves step by step across the states of the Louisville environment. The state, \mathbf{X}_t , is observed as soon as an environment-changing event fires, i.e., a trip request is scheduled; such observation is available to P and D , which will consequently pick an action, a_t . The simulator will then

3.3.2. Agents architecture

The agents are Deep-Q-Networks (DQN) implementing an ϵ -greedy policy with experience replay [14] belonging to the family of Q-learning. It is an off-policy approach towards deep RL wherein the agent estimates the expected reward for future actions from a given state without following an actual greedy policy, but instead relying on a behaviour policy enriched from direct experience with the environment to update the online policy, by satisfying Bellman's optimality equation. Such an approach is better suited for large state spaces, \mathcal{S} , against rather limited action spaces, \mathcal{A} , which we will see to be our case. In fact, they are Rainbow agents [15], a state-of-the-art DQN agents, which we have found beneficial for the three following main features: double Q-learning helps in preventing overestimation of the action values which may lead to very unpleasant proposals; distributional Q-learning helps in investigating the importance of the value distribution, which we find necessary to achieve long-term balance of the ESB-DQN system; prioritized experience replay helps in selecting the subset of previously experienced observations that are the most relevant, which we find necessary to characterize the complexity of the dynamics behind a free-floating sharing system .

Both the pick-up agent and the drop-off agent comprise a funnel-like three-layer fully connected network with ReLU activation functions, whose role is to flatten the input and extract a latent representation as a single vector of 256 units. The input of the network is the last observed environment state, \mathbf{X}_t , whereas the output feeds the standard Rainbow network that produces a distribution of logits, whose maximum value identifies the action picked by each agent, $a_{P,t}$ and $a_{D,t}$, respectively. The action space is limited to 9 different choices, corresponding to the 8 cardinal directions mapping the 8 adjacent zones (i.e., 1-hop neighbourhood) plus the calling zone, $p(i)$ or $d(i)$ respectively, which function as the NOP actions.

Following this formulation of the action space, and recalling Figure 1, it becomes clear why the ESB-DQN environment is constrained by a large set of invalid actions. In fact, in the early stages of the RL agents life-cycle, the expectation of picking an invalid action from any given zone z at any given time t far exceeds its complementary, which is further evidence of the need for outer aid for the RL agents to well characterize the dynamics of the system.

3.3.3. Reward

The following functions are defined to compute the reward:

$$\omega(z, t) = \Delta(z, t) \exp \left[- \left(\frac{1}{d(z, t)^+} N_A(z, t)^+ \right)^{\text{sign}(\Delta(z, t))} \right] \quad (2)$$

$$\psi(z, t) = N_D(z, t) \exp \left(- \frac{1}{d(z, t)^+} N_A(z, t) \right) \quad (3)$$

where:

- $\Delta(z, t)$: expected deficit of e-scooters at zone z at time t ;
- $d(z, t)$: future demand of e-scooters in z at time t (in a time interval $t + \Delta t$);

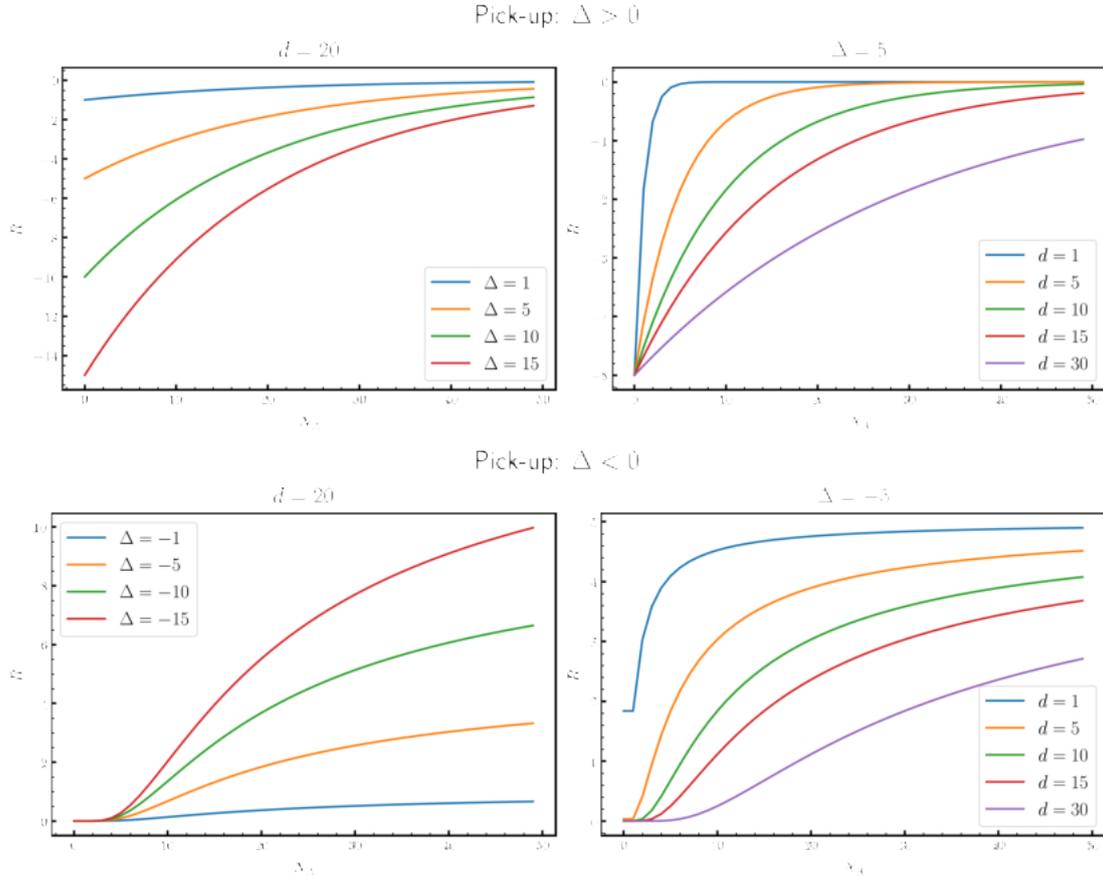


Figure 2: Pick-up action: some examples of the reward function R_P for different values of the parameters Δ, d . **(Upper row)** Negative reward: the agent suggests to pick-up a vehicle from a zone having an expected deficit of vehicles ($\Delta > 0$). As Δ and d increase, the reward is smaller because the expected deficit condition will be worsened. In both cases, the larger the number of available vehicles the higher the curve, as the deficit condition will be alleviated. **(Bottom row)** Positive reward: the agent suggests to pick-up a vehicle from a zone having an expected surplus of vehicles ($\Delta < 0$). The larger Δ the larger the reward, reflecting how problematic the surplus being improved. Similarly, when the number of available vehicles is high, the rebalancing effect is considered more valuable. As the demand d increases, the reward decreases because the pick-up may negatively affect the long-term balance of the zone.

- $N_A(z, t)$: number of available e-scooters in z at time t ;
- $N_D(z, t)$: number of dead e-scooters in z at time t ;
- $(\cdot)^+$ denotes the function $\max(\cdot, 1)$ and is used to prevent from division by zero.

Drop-off agent. Let $\hat{d}(i)$ be the chosen alternative drop-off zone for trip i at time t , $\mathcal{N}_{\hat{d}(i)}$ be the set of valid neighbours around $\hat{d}(i)$. If the state of charge of the vehicle s at the end of the trip is greater than the battery swap threshold, i.e. $b(s) - e_i \geq \alpha C$, then the reward

corresponding to each of the alternative zones $z \in \mathcal{N}_{\hat{d}(i)}$ is:

$$R_D(z, t) = \begin{cases} \omega(z, t) & \text{if } \hat{d}(i) = z \\ -\omega(z, t) & \text{otherwise} \end{cases}$$

Otherwise:

$$R_D(z, t) = \begin{cases} \psi(z, t) & \text{if } \hat{d}(i) = z \\ -\psi(z, t) & \text{otherwise} \end{cases}$$

The overall reward for the choice $\hat{d}(i)$ is:

$$\bar{R}_D(\hat{d}(i), t) = \begin{cases} \frac{1}{|\mathcal{N}_{\hat{d}(i)}|} \sum_{z \in \mathcal{N}_{\hat{d}(i)}} R_D(z, t) & \text{if drop-off action is valid} \\ -\gamma_D \max_{z \in \mathcal{N}_{\hat{d}(i)}} |R_D(z, t)| & \text{otherwise} \end{cases}$$

with γ_D being a constant which modulates the penalty of an invalid drop-off action.

Pick-up agent. Let $\hat{p}(i)$ be the chosen alternative pick-up zone for trip i at time t , $\mathcal{N}_{\hat{p}(i)}$ be the set of valid neighbours around $\hat{p}(i)$. The reward corresponding to each of the zones $z \in \mathcal{N}_{\hat{p}(i)}$ is:

$$R_P(z, t) = \begin{cases} -\omega(z, t) & \text{if } \hat{p}(i) = z \\ \omega(z, t) & \text{otherwise} \end{cases}$$

The overall reward for the choice $\hat{p}(i)$ is:

$$\bar{R}_P(\hat{p}(i), t) = \begin{cases} \frac{1}{|\mathcal{N}_{\hat{p}(i)}|} \sum_{z \in \mathcal{N}_{\hat{p}(i)}} R_P(z, t) & \text{if pick-up action is valid} \\ -\gamma_P \max_{z \in \mathcal{N}_{\hat{p}(i)}} |R_P(z, t)| & \text{otherwise} \end{cases}$$

with γ_P being a constant which modulates the penalty of an invalid pick-up action. Figure 2 shows some examples of the reward function R_P for different values of the parameters Δ , d .

3.4. Lives mechanism

As we have anticipated in Section 3.3, a major role during the training of the ESB-DQN system has been played by the parameter regarding the number of lives, k . The continuity of the simulation is a key factor for the eventual learning of the RL agents, as interrupting the simulation to just start it over too often, as soon as an invalid action happens, would slow down the convergence by a considerable margin, given how full of potential invalid actions ESB-DQN environment is.

To overcome this limitation, we have borrowed the concept of lives from Atari: every time one of the two agents or both commit an invalid action, the whole environment loses a life. By doing so, an invalid action does not immediately lead to a terminal state, but takes it closer to the ESB-DQN state. On life loss, the discount for the timestep t is zeroed, cancelling any connection between the previous and later events, and the agents are set to perform a NOP.

Let $a_t = (a_{t,P}, a_{t,D})$ be the generic action for the simulator taken at time t , defined as the resulting combination of the action picked by the pick-up (P) agent, $a_{P,t}$, and the action picked by the drop-off (D) agent, $a_{D,t}$. The set of invalid actions has been set as follows:

- either zone corresponding to $a_{P,t}$ or $a_{D,t}$ is invalid: $z_{P,t} \notin Z \cup z_{D,t} \notin Z$, with Z the set of valid zones of the city of Louisville;
- the zones corresponding to $a_{P,t}$ and $a_{D,t}$ are equal: $z_{P,t} = z_{D,t}$;
- the zones corresponding to $a_{P,t}$ and $a_{D,t}$ are equal to the original zone of opposite type: $z_{P,t} = \hat{z}_{D,t} \cup z_{D,t} = \hat{z}_{P,t}$;
- the original zones $\hat{z}_{P,t}$ and $\hat{z}_{D,t}$ are equal: $\hat{z}_{P,t} = \hat{z}_{D,t}$;
- the suggested pick-up zone does not have a suitable vehicle ready: $V_{P,\text{avail}} = \emptyset$

As soon as k reaches 0, then the simulation is stopped. Indeed, we would not want our RL agents to learn the dynamics of the environment while committing thousands of errors. It is important to note that by implying the concept of lives, the training framework of RL agents has turned into a sort of collaborative RL framework, wherein both P and D cannot rely solely on their capabilities to reach the goal, but even on the other's to reach a common goal: if D was to lose a life, P would lose it as well, and viceversa.

4. Experiments and Results

The ESB-DQN system has been trained to learn the best alternative zone proposals throughout simulations with the Louisville dataset. The aim of the experiments has been to evaluate whether incentivizing users to pick-up/drop-off vehicles in alternative zones can preserve a good quality of service with a reduced number of relocation and battery swap workers.

The quality of service is measured through the satisfied demand D_{sat} , defined as follows:

$$D_{\text{sat}} = \frac{N_{\text{trips}} - N_{\text{unsat}}}{N_{\text{trips}}} \quad (4)$$

where N_{trips} is the total number of trips, N_{unsat} is the number of *unsatisfied* trips (no available vehicles in the pick-up zone and in the 1-hop neighbourhood), both measured over a given fixed time interval T_{sim} . Through all our experiments, T_{sim} is equal to 1 day.

The parameters of the simulator have been set as follows:

- the number of available e-scooters is $|\mathcal{S}| = 400$;
- the size of the zones is $l = 200m^2$;
- the battery capacity is $B = 425 \text{ Wh}$ with $\alpha = 0.3$, whereas the energy required to complete a trip is proportional to the driving distance by a factor of 11 Wh/km (as suggested in [4]);
- the user willingness is $w = 1$;
- the battery swap operations are scheduled every $T_s = 1\text{h}$;
- the relocation operations are scheduled every $T_r = 1\text{h}$ in a working time interval $T_{\text{work}} = [9\text{AM}-6\text{PM}]$.

The fleet size $|\mathcal{S}|$ and the user willingness w immediately stand out from the lot of parameters. The former has been set to half the nominal fleet size granted by the city of Louisville. Indeed, as further experiments on cities with more complex dynamics have not been conducted for the time being, we have decided to restrict Louisville to a worst case scenario, as the quality of service would remain strong nonetheless (88%). The latter has been set to 1, as in the training phase we wanted to let both RL agents experience as much of the environment as possible, regardless of whether they would be actually asked to do so.

The parameters of the reinforcement learning system have been set as follows:

- the optimizer is Adam with a learning rate of 6.25×10^{-5} ;
- the learning period is 16;
- the batch size is 32;
- the timesteps are aggregated to look back to the last 3 timesteps before any decision process takes place;
- the global gradient norm clipping is 10;
- the importance sampling exponent ranges in $[0.4, 1]$;
- the experience replay buffer has size 5.2×10^3 , amounting to almost 30 full repetitions of the same day over and over, with priority exponent of 0.5;
- the target network update period is 1.6×10^2 ;
- the number of iterations is 48;
- the number of trips per episode is 1.3×10^3 , amounting to almost 10 full repetitions of the same day over and over;
- the number of validation trips is 2.6×10^3 ;
- the number of training trips is 5.2×10^3 ;
- the number of total lives k has been set to 100.

Moreover, concerning the reward function, the future demand is computed in a time interval $\Delta t = 1\text{h}$, whereas the constants modulating the penalties are $\gamma_D = \gamma_P = 2$. Also, every 3 iterations a checkpoint has been stored locally for evaluation purposes.

In the first experiment, the model has been trained from scratch with the number of relocation workers being $n_{\text{swap}} = 12$ and the number of battery swap workers being $n_{\text{rel}} = 6$. Other two experiments have been performed, by drastically reducing the number of workers and applying transfer learning from the pre-trained P and D agents. In particular, in the second experiment we have fixed $n_{\text{swap}} = 6$, $n_{\text{rel}} = 3$ and in the third experiment $n_{\text{swap}} = n_{\text{rel}} = 1$.

The final results in validation are shown in Table 2. The evolution of the satisfied demand during the learning procedure is represented in Figure 3.

The number of validation/training trips follows DeepMind’s suggested ratio of 1 : 2 between the online and the offline ϵ -greedy networks. For example, if a training episode would experience 1000 trips, a validation episode would experience only half of those. A single iteration took over 1 hour on a PC equipped with a GeForce GTX 1650 Ti GPU with 4GB of memory along with an Intel i7-10750H CPU with 32GB of RAM. Both CPU and GPU specs are crucial, as the SimPy processes undergoing the simulation run solely on CPU, whereas the forward and backward

pass of the RL agents' networks happen on GPU.

As shown in Figure 3(a), the two agents trained from scratch cause a decrease in the satisfied demand during the first iterations, due to their random behaviour with no previous experience. After around 25 iterations their policies have been efficiently updated. The level of satisfied demand has improved with respect to the baseline - referred to a standard mobility service with no user incentives. More interestingly, by reducing the number of workers and applying transfer learning, it is possible to observe again a beneficial effect over the satisfied demand. In particular, Figure 3(c) shows that in the critical scenario with $n_{\text{swap}} = n_{\text{rel}} = 1$ the satisfied demand is constantly larger with respect to the baseline. This means that by following the proposal of alternative pick-up and drop-off zones, users are actively participating to the system rebalancing and contributing to a positive increase of the quality of service.

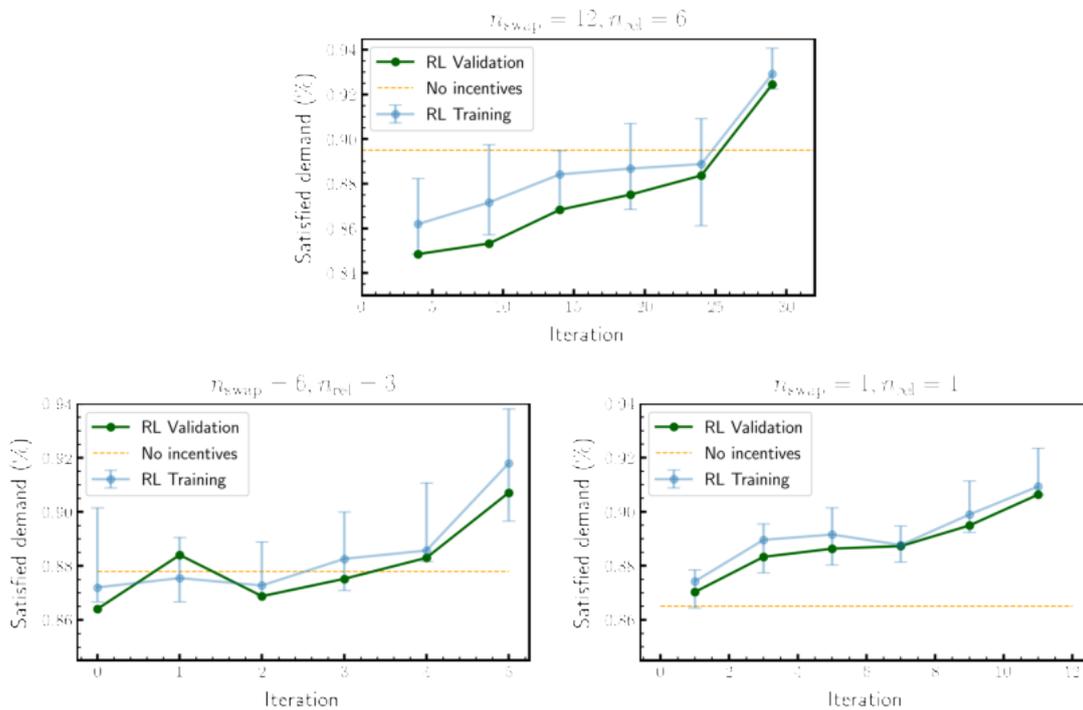


Figure 3: Evaluation of the satisfied demand during the learning procedure for training and validation agents versus a baseline model with no incentive policy (user willingness $w = 0$). The value of the parameters is shown in the titles. (a) Model trained from scratch, (b), (c) Transfer learning.

5. Discussion and Future Works

In this paper, we presented ESB-DQN, a multi-agent system based on deep reinforcement learning able to interact with a simulator in order to learn alternative pick-up and drop-off zones in e-scooter sharing services. The main objective is to combat the imbalance problem

Table 2
Results

Parameters	Training mode	N iterations	Satisfied demand D_{sat}	
			ESB-DQN	No incentives
$n_{\text{swap}} = 12, n_{\text{rel}} = 6$	from scratch	30	0.92	0.89
$n_{\text{swap}} = 6, n_{\text{rel}} = 3$	transfer learning	6	0.91	0.88
$n_{\text{swap}} = 1, n_{\text{rel}} = 1$	transfer learning	12	0.90	0.86

by providing user incentives in order to optimize vehicle availability as well as battery swap and relocation operations. At present, ESB-DQN expects to know the original pick-up and drop-off locations of each generic scheduled trip, $p(i)$ and $d(i)$, beforehand, in order to produce proper suggestions. Of course, such a constraint poses a strong limitation to the effectiveness of the system, as it is impractical to always expect users to know their future drop-off location before initiating the trip. Nevertheless, following the way the original simulator handles the notion of booking requests as pairs of pick-up and drop-off locations, forcing both pick-up and drop-off agents, P and D , to operate synchronously was a necessary starting point. The natural evolution of the ESB-DQN system requires the untying of this synchrony, to let P and D affect the state of the environment independently at different stages.

Preliminary experiments on real e-scooter data from Louisville (US) have shown encouraging results on the satisfied demand of the system, even with a strongly reduced number of workers.

Further experiments are required for a comprehensive evaluation of the ESB-DQN system. By varying different parameters of the simulator, e.g., the number of e-scooters $|S|$, the number of relocation workers n_{rel} or battery swap workers n_{swap} , it is possible to study how each of them, in turn, affects the user incentives policy. It is worth mentioning that more accurate demand forecasts for the computation of $\delta(z, t)$ in Eq. 2, 3 can be adopted with the aim of getting further improvements on the overall performance of the ESB-DQN system.

A fundamental effort should be devoted to scale-up experiments on a larger temporal scale and on larger datasets (e.g., Austin open data [16]). A larger number of iterations would indeed reflect in a better characterisation of the ϵ -greedy policy. Indeed, despite both RL agents have reached some sort of convergence with even a few iterations, there may be a few specific corner cases of states that leave them both unable to decide with high consistency. Concerning a possible speed-up, since SimPy processes run on the CPU, there is a lot of time left to gain by optimizing the underlying simulator to fasten the run time of a single day. On the other hand, the code related to the multi-agent system is already optimized for GPUs and TPUs.

Another interesting possibility is to apply the ESB-DQN system to other mobility sharing systems with different vehicles (e.g., e-bikes, e-moped). Provided with the right data and the proper scenario parameters (e.g., fuel type, fuel consumption, maintenance costs) both the simulator and the multi-agent system can be directly applied to such problems.

The proposed approach may be deployed in real mobility systems as a real-time service following the REST paradigm, integrated into existing app used by mobility service providers. A prototype of the API is under development along with a chatbot intended to provide a natural language interface the users could interact with as well.

References

- [1] A. Chang, L. Miranda-Moreno, R. Clewlow, L. Sun, Trend or Fad? Deciphering the Enablers of Micromobility in the US, Technical Report, 2019.
- [2] C. S. Smith, J. P. Schwieterman, E-scooter Scenarios: Evaluating the Potential Mobility Benefits of Shared Dockless Scooters in Chicago (2018).
- [3] R. Zhu, X. Zhang, D. Kondor, P. Santi, C. Ratti, Understanding Spatio-temporal Heterogeneity of Bike-sharing and Scooter-sharing Mobility, *Computers, Environment and Urban Systems* 81 (2020) 101483.
- [4] A. Ciociola, M. Cocca, D. Giordano, L. Vassio, M. Mellia, E-scooter Sharing: Leveraging Open Data for System Design, in: 2020 IEEE/ACM DS-RT, IEEE, 2020, pp. 1–8.
- [5] J. Pfrommer, J. Warrington, G. Schildbach, M. Morari, Dynamic Vehicle Redistribution and Online Price Incentives in Shared Mobility Systems, *IEEE Transactions on Intelligent Transportation Systems* 15 (2014) 1567–1578. doi:10.1109/TITS.2014.2303986.
- [6] C. Fricker, N. Gast, Incentives and Redistribution in Homogeneous Bike-sharing Systems with Stations of Finite Capacity, *EURO Journal on Transportation and Logistics* 5 (2016) 261–291. URL: <https://www.sciencedirect.com/science/article/pii/S2192437620300959>. doi:<https://doi.org/10.1007/s13676-014-0053-5>.
- [7] P. Yi, F. Huang, J. Peng, A Rebalancing Strategy for the Imbalance Problem in Bike-sharing Systems, *Energies* 12 (2019). URL: <https://www.mdpi.com/1996-1073/12/13/2578>. doi:10.3390/en12132578.
- [8] R. Regue, W. Recker, Proactive Vehicle Routing with Inferred Demand to Solve the Bike-sharing Rebalancing Problem, *Transportation Research Part E: Logistics and Transportation Review* 72 (2014) 192–209. URL: <https://www.sciencedirect.com/science/article/pii/S1366554514001720>. doi:<https://doi.org/10.1016/j.tre.2014.10.005>.
- [9] J. Liu, L. Sun, W. Chen, H. Xiong, Rebalancing Bike-sharing Systems: A Multi-source Data Smart Optimization, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 1005–1014. URL: <https://doi.org/10.1145/2939672.2939776>. doi:10.1145/2939672.2939776.
- [10] Y. Duan, J. Wu, Optimizing Rebalance Scheme for Dock-less Bike-sharing Systems with Adaptive User Incentive, in: 2019 IEEE MDM, IEEE, 2019, pp. 176–181.
- [11] L. Pan, Q. Cai, other authors, A Deep Reinforcement Learning Framework for Rebalancing Dockless Bike-sharing Systems, in: Proc. of AAAI, volume 33, 2019, pp. 1393–1400.
- [12] DeepMind, DQN Zoo, 2020. URL: https://github.com/deepmind/dqn_zoo.
- [13] H. Spieker, Constraint-guided Reinforcement Learning: Augmenting the Agent-Environment Interaction, *CoRR abs/2104.11918* (2021). arXiv:2104.11918.
- [14] V. Mnih, K. Kavukcuoglu, other authors, Human-level Control through Deep Reinforcement Learning, *Nature* 518 (2015) 529–533.
- [15] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. G. Azar, D. Silver, Rainbow: Combining Improvements in Deep Reinforcement Learning, *CoRR abs/1710.02298* (2017). URL: <http://arxiv.org/abs/1710.02298>. arXiv:1710.02298.
- [16] Austin Shared Micro-mobility Open Data, 2019. URL: <https://dev.socrata.com/foundry/data.austintexas.gov/7d8e-dm7r>.

Graph Neural Networks as the Copula Mundi between Logic and Machine Learning: a Roadmap

Andrea Agiollo, Giovanni Ciatto and Andrea Omicini

Dipartimento di Informatica – Scienza e Ingegneria (DISI), ALMA MATER STUDIORUM—Università di Bologna, Italy

Abstract

Combining machine learning (ML) and computational logic (CL) is hard, mostly because of the inherently-different ways they use to represent knowledge. In fact, while ML relies on fixed-size numeric representations leveraging on vectors, matrices, or tensors of real numbers, CL relies on logic terms and clauses—which are unlimited in size and structure.

Graph neural networks (GNN) are a novelty in the ML world introduced for dealing with graph-structured data in a sub-symbolic way. In other words, GNN pave the way towards the application of ML to logic clauses and knowledge bases. However, there are several ways to encode logic knowledge into graphs: which is the best one heavily depends on the specific task at hand.

Accordingly, in this paper, we (i) elicit a number of problems from the field of CL that may benefit from many graph-related problems where GNN has been proved effective; (ii) exemplify the application of GNN to logic theories via an end-to-end toy example, to demonstrate the many intricacies hidden behind the technique; (iii) discuss the possible future directions of the application of GNN to CL in general, pointing out opportunities and open issues.

Keywords

graph neural networks, machine learning, embedding, computational logic,

1. Introduction

Artificial Intelligence (AI) has gained significant importance in our ever evolving and technology-focused world. Rising popularity for this field can be attributed to the overwhelming success of sub-symbolic techniques like deep learning. However, along with AI increase in popularity, there exists public concern related to the relevant role that intelligent systems will bear in human society, in particular for the lack of understandability of AI systems.

Whenever intelligent systems play critical roles within human society, they should be made clearly understandable from a human perspective. This need has been taken under deep consideration by the XAI (eXplainable Artificial Intelligence [1]) research community. XAI approaches would seemingly require the integration between successful sub-symbolic techniques and symbolic frameworks in order to reach their main goals [2]. Among the symbolic approaches under scrutiny nowadays, logic-based techniques possibly represent the most straightforward path towards human understanding. The reason behind that is straightforward: symbols are far

WOA 2021: Workshop “From Objects to Agents”, September 1–3, 2021, Bologna, Italy

✉ andrea.agiollo@unibo.it (A. Agiollo); giovanni.ciatto@unibo.it (G. Ciatto); andrea.omicini@unibo.it (A. Omicini)

ORCID 0000-0003-0531-1978 (A. Agiollo); 0000-0002-1841-8996 (G. Ciatto); 0000-0002-6655-3869 (A. Omicini)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

closer to the way our conscious mind works than the vectors, tensors, and algebraic operations sub-symbolic AI is built upon. Along this line, it is essential for the future of AI to harmonise and integrate symbolic – and, in particular, logic-based – and sub-symbolic AI—and, in particular, neural networks.

The idea that symbolic and sub-symbolic AI are complementary under a number of dimensions is well understood [3, 4]. Few significant works have been proposed integrating neural networks with fuzzy logic [5, 6]. However, a general purpose solution to the problem is still lacking. Should we speculate on what is bringing inertia into this field of research, we would argue that the two approaches to AI are fundamentally dual w.r.t. the way they represent information—in short, formulæ vs. tensors. Indeed, a basic requirement of any integrated system involving both symbolic and sub-symbolic processing is the capability to either convert symbols into tensors, or vice versa—possibly both. This problem is hard to formalise in the general case.

In this work we focus on the simpler problem of enabling the sub-symbolic processing of logic knowledge-bases. In particular, we focus on the exploitation of neural networks (NN) as a means to complement computational logic (CL) when it comes to process symbolic data expressed in logic form. NN have indeed proven their strengths in many scenarios, ranging from computer vision to natural language processing, and, more recently, on knowledge graphs and graph-like data.

Accordingly, in this work we study the possible use of graphs as a bridge between CL – among the most prominent branches in symbolic AI – and neural networks—among the most flexible and successful approaches in ML. We consider graphs as the ideal bridge because of their versatility in representing virtually any sort of data structures, there including recursive ones—an aspect that is very common in logics as well as quite critical in ML.

Accordingly, we investigate the potential and the intricacies of blending CL and graph neural networks (GNN) [7]—a novel category of NN which is particularly well-suited to handle graph-like data. In particular, the aim of the work is to understand how and to what extent CL systems can benefit from GNN and graph-oriented ML in general. Along this line, the contributions brought about by this work are the following: *(i)* we categorise a number of problems from the field of CL which may take advantage from GNN, given the existence of a clear way to express the problem in terms of graphs; *(ii)* we discuss a number of graph-based tasks which can be suitably tackled via GNN, and discuss how the aforementioned problems can be mapped into such existing tasks; *(iii)* we present an end-to-end scenario where GNN are applied to a simple logic knowledge base to detect missing facts link prediction: the toy example is used to demonstrate the many intricacies hidden behind the application of GNN over logic data; *(iv)* finally, we provide a research roadmap that researchers interested in this field may follow for future works, eliciting opportunities arising from the integration between CL and GNN.

2. State of the Art

In this section we briefly introduce graph neural networks, along with the tasks these aim at solving and their working principle (Section 2.1). Then we give an overall background on computational logic, focussing in particular on the many ways to translate a knowledge base into a graph—in order to make it processable via GNN (Section 2.2).

2.1. Graph Neural Networks

In recent years, machine (ML) and deep learning (DL) techniques have disrupted the way complex data-driven tasks – ranging from image classification to speech recognition, natural language processing and many more – are tackled. However, most ML approaches can handle data having a *fixed* structure and size—most notably, vectors, matrices, or tensors of real numbers. This may be troublesome in some contexts, given the ever-increasing popularity of applications involving data which cannot be suitably represented by fixed-size, rigid structures. Among the most relevant applications in this category, we can find a number of *graph*-processing scenarios. To tackle this issue, research effort has focused on extending ML approaches to graph-structured data. Notably, *graph neural networks* (GNN) are a novel approach to let ordinary NN-based processing be applied to graphs.

GNN are mathematical models operating upon *directed* graphs, whose vertices (resp., arcs) are labelled with vectors (or matrices, or tensors) of real numbers, each one carrying further numeric information about the corresponding vertex (resp., arc). GNN output depends on the learning task to be performed, which commonly ranging in any of three wide classes of tasks: (i) the classification of similar graphs having different topology – i.e. *graph classification* – [8], (ii) the classification of vertices of unknown graphs – i.e. *nodes classification* – [9], or (iii) the identification of missing but statistically probable arcs—i.e. *link prediction* [10].

Graphs handled by GNN usually carry information in the form of vertices and arcs vectors/matrices/tensors. Consider for instance the graph representation of a chemical molecule: it is necessary to represent the sort of atomic element associated with each vertex. The same holds for the details of the chemical bonds among two any atoms of a molecule—which must be associated with the graph’s arcs.

Accordingly, we here consider vertices of a graph to be characterised by specific features called vertex attributes, represented as vectors of the form $\mathbf{x}_v \in \mathbb{R}^d$, where v enumerates the vertices of a graph, and d is the dimension of all vectors of all vertices. We also assume the existence of total ordering among the vertices in V , so we may refer to a vertex by its index i . We denote by $\mathbf{X} \in \mathbb{R}^{n \times d}$ the matrix of all vertex attributes, attained concatenating each vector \mathbf{x}_v along a single dimension. There, n is the number of vertices in G . We also denote by $N(v)$ the neighbourhood of a vertex v , here considered as the set of all vertices u directly linked to vertex v by an outgoing arc, i.e. $N(v) = \{u \in V \mid (v, u) \in E\}$.

Concerning arcs, we denote by $a_{v,w} \equiv (v, w) \in A$ the arc connecting vertex v to vertex w . Similarly to vertices, we consider arcs in the graph as characterised by specific arc attributes, represented by vectors of the form $\mathbf{a}_{v,w} \in \mathbb{R}^c$, where c is the cardinality of the all vectors of all arcs. Finally, we denote by $\mathbf{A} \in \mathbb{R}^{m \times c}$ matrix containing all arc attributes. There, m is the number of arcs in G .

Figure 1 depicts the general architecture of a GNN. It consists of a cascade of three functional blocks (each one composed by one or more layers of neurons) serving specific purposes:

Convolutor. The first block of the GNN is in charge of accepting the graph G as input and producing a new *convoluted* graph G' as output, having the same topology of G , where the vector associated with each vertex v has been replaced by another vector describing the relevance of each vertex w.r.t. the whole graph G . Convolutor block relies on convolution

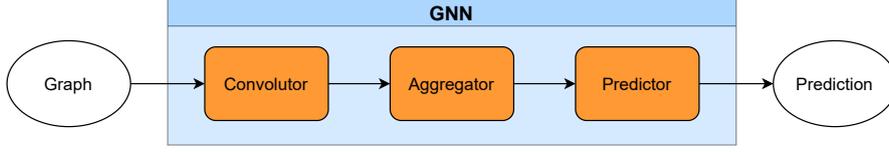


Figure 1: Graph Neural Networks are composed as a cascade of simpler blocks.

operation, extensively exploited in DL to express relevance of local data w.r.t. to global data.

The application of convolution operation to non-Euclidean data – like graphs – is not straightforward. An equivalent notion of convolution over graphs has been proposed to compute the relevance of each vertex w.r.t. to its neighbours. Graph convolution is defined over a single vertex v and its neighbourhood $N(v)$ and relies on three successive phases:

propagation – the information $\mathbf{x}_{v'}$ belonging to each vertex $v' \in N(v)$ is weighted by the information $\mathbf{a}_{v,v'}$ belonging to the arc among v and v' and then propagated to vertex v ;

aggregation – the information propagated from each vertex $v' \in N(v)$ to v is aggregated using a *parametric* aggregation function;

transformation – the aggregated information corresponding to vertex v is transformed into a new embedding vector and assigned back to vertex v , as its new state \mathbf{x}'_v .

The single convolution operation is applied in parallel to each vertex in G .

Inside the convulator block, the graph convolution procedure is repeated T times. The overall effect of step t is the production of a new graph G^t having the same vertices and arcs of G , where the i^{th} vertex at step $t + 1$ carries a more convoluted information than the same vertex at step t . More formally, the relation tying each layer of the convulator block with its successor is captured by the following recursive equation:

$$\mathbf{x}_v^{t+1} = \Theta^t \left(\mathbf{x}_v^t, \uplus_{w \in N(v)} \Xi^t (\mathbf{x}_v^t, \mathbf{x}_w^t, \mathbf{a}_{v,w}) \right) \quad (1)$$

where functions Ξ^t , \uplus , and Θ^t represent the *propagation*, *aggregation*, and *transformation* phases respectively.

Function Ξ^t , in particular, propagates the information belonging to all neighbours $w \in N(v)$ of vertex v through the arc that connects the two. This function must be differentiable and parametric – to be amenable of optimisation through the back-propagation algorithm [11] –, other than layer-specific and shared among all vertices of the graph.

Function \uplus aims at aggregating the information received by each vertex v from its neighbourhood. For this reason, it must be variadic and permutation invariant, other than being shared among all layers and all vertices of the graph.

Finally, Θ^t is a differentiable, parametric, and layer-specific function, aimed at aggregating neighbourhood information to compute the vertex attributes for vertex v at step t .

Aggregator. The convoluted graph G' is passed to an aggregator block that produces a fixed-sized representation of the graph G' —called *embedding* of G ;

Predictor. The embedding produced by the aggregator block, being fixed in size, can be used as the input of an ordinary NN – namely, the predictor block – to solve ordinary ML tasks (e.g., classification or regression) on the original graph G .

2.2. Logic Theories as Graphs

Computational Logics (CL) essentially deals with logics as a means for computing [12]. Provided that knowledge can be expressed in terms of logic theories (a.k.a. knowledge bases, KB), made up of several logic clauses, CL endows software agents with automated reasoning capabilities, via many sorts of *inference* rules.

Knowledge bases can be encoded into graphs in several ways and to serve disparate purposes. Generally speaking, KB can be encoded into graphs by aggregating the graphs attained by encoding all clause therein contained. In all such cases, encoding schemas can act at either the *semantic* or at the *syntactic* level.

Encoding schemas operating at the *syntactic* level capture static relationships inferable from the mere syntax of clauses and KB. Abstract syntax trees (AST) are the simplest example of graphs which can be attained from KB. They consist of direct acyclic graphs where vertices are of as many sorts as the possible syntactical categories of which may occur in a KB – namely, theories, clauses, predicates, or terms –, whereas arcs simply describe container-contained relations among vertices. Dependency graphs are another kind of graph that may be attained from a KB. They consist of directed graphs where each vertex represents a predicate, and each arc represents a logic dependency among two predicates—meaning that the predicate corresponding to the destination vertex must be proven true before the predicate corresponding to the source vertex, in a resolution process.

Encoding schemas operating at the *semantic* level capture high level relationships that can be inferred from the actual meaning of a logic theory. Entity-Relationship (ER) graphs are the simplest kind of graph in this category. They aim at expressing via graphs the same information a ground KB expresses via formulæ. They consist of directed graphs where vertices may either represent entities (i.e. terms) or relationships (i.e. predicates) and arcs represent the participation of an entity into a relationship. Triplet graphs are another simple way of representing ground theories where all terms are constants and all predicates are either unary or binary. When this is the case, each constant is considered an entity, binary predicates are considered as relations among two entities, and unary predicates are considered as properties an entity may or may not have. Thus, a graph can be attained by defining a vertex for each different constant in a KB, and arc for each couple of constants involved in at least a binary predicate.

3. Processing Logic Knowledge via GNN

In this section we present a research roadmap eliciting the potential bridges among CL and GNN. We first identify four relevant tasks from CL where, we believe, sub-symbolic processing may have a role to play. We then discuss how all such tasks can be mapped into as many well-known graph manipulation tasks, for which GNN have already been exploited. Finally, we present a general framework for sub-symbolically processing logic information via GNN, and we elicit the many constraints a designer should satisfy when doing so.

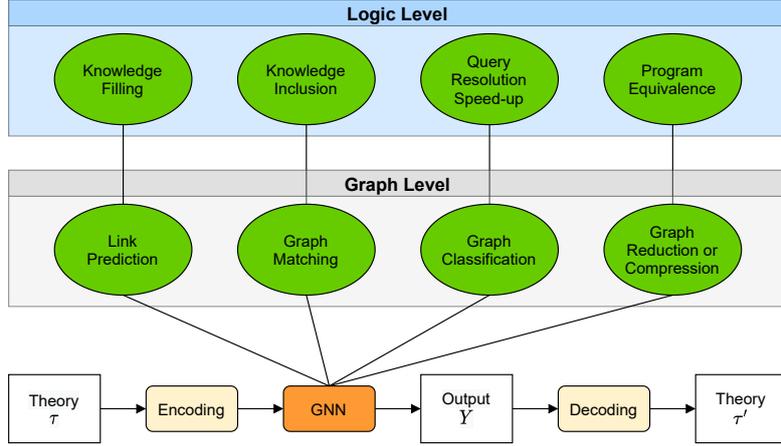


Figure 2: Proposed roadmap along with its theoretical framework. Logic level and graph level can be mapped directly, in order for logical problems to benefit from sub-symbolical techniques – e.g. GNN – available at the graph level.

3.1. Logical Tasks

Manipulation of logic knowledge enables the resolution of complex queries via logical inference. There exist, however, relevant tasks which are hard to formalise or solve into the logic realm, because of either their numerical nature or algorithmic infeasibility. Here, in particular, we identify four relevant operations on knowledge bases for which, we argue, it is worth investigating sub-symbolic solutions.

The tasks considered – shown in the upper box of Figure 2 – are (i) knowledge filling, (ii) knowledge inclusion, (iii) program equivalence, and (iv) resolution speed-up.

Knowledge Filling. Entities and relations available in a logic theory may sometime lack some instances. For example, this may happen because the human operator handcrafting the theory was imprecise or when an agent’s knowledge is incomplete. When this is the case, we consider the knowledge base as *fragmented*.

To deal with such fragmented theories, it may be useful to identify missing relations between existing entities. This task may be tackled via statistical analysis of the theory under examination, which may lead to the identification of latent relations among entities.

A knowledge filling problem would be hard to handle symbolically, as logic reasoners commonly struggle in processing knowledge they do not have. While most solvers operate under a closed world assumption – letting them considers as false everything they do not explicitly know to be true –, even the ones operating under an open world assumption do not commonly include mechanisms to generate new knowledge out of thin air. In all such cases, the coherence and completeness of the knowledge base is usually considered as an a-priori requirement for logic computations to work properly. Conversely, in the sub-symbolic realm, semantic similarities among the entities and relations of a logic theory may be better captured, which may help reconstructing missing facts.

Consider for instance the case of a simple theory representing kinship relationships. The

lack of a single relation – say that “John and Mary are siblings” – may significantly hinder a solver’s ability to deduce kinships among entities of a family—e.g. “the sons of John and Mary are cousins”. The solution for this task is not straightforward, thus attracting our attention.

Knowledge Inclusion. Knowledge inclusion represents the task checking whether a given theory (usually smaller) is complementary w.r.t. another given theory (usually larger) or not. The same clauses could occur with slightly-different shapes—e.g. using different predicates/functor names or different positions arguments in the same predicates.

This task requires the ability to express equivalence or similarity among *groups* of clauses, which is not straightforward [13, 14]. Computing exact solutions to this problem may soon become infeasible as the dimensions of the involved theories increases. Conversely, in the sub-symbolic realm, the same problem may be modelled as a pattern-matching problem. This may pave the way towards the computation of *approximate* solutions to the knowledge inclusion problem in reasonable time.

As an example, consider multiple agents sharing partially-similar information. In this case, it would be desirable to identify agents common knowledge and ease their interaction. Suppose that agents information is expressed via two theories τ_1 and τ_2 , both representing family trees. τ_1 expresses 1st degree relatives only, while τ_2 includes also 2nd degree relatives. τ_1 may consider more/less/different family members w.r.t. τ_2 , and kinships may also be defined in different ways between the two theories. However, τ_1 is – logically speaking – a subset of τ_2 , and we need to detect this property.

Program Equivalence. Program equivalence represents the task of computing a simpler and equivalent theory τ' starting from a theory τ . This may imply removing redundancies and simplifying clauses. As for the knowledge inclusion task, program equivalence requires a procedure to compare sets of clauses, other than the capability of generating reduced equivalent variants of clauses. It is our opinion that both these procedures may be better expressed into sub-symbolical realm.

Considering again agents storing kinships information, it may be desirable to compress a single agent information to produce a new theory for a simpler agent. This new theory should ideally have fewer rules, while spanning the same family tree of the original theory. Such a requirement is difficult to satisfy, and would probably require notions of semantically-equivalent sets of kinships—e.g., the set of relations containing only *parent* spans the same family tree of the set of relations $\{mother, father\}$.

Query Resolution Speed-Up. Logic theories are commonly exploited by logic solvers to draw inferences, via some resolution procedure. The execution time of any query resolution vastly depends on the complexity of the algorithm(s) expressed by the logic theory. To this regard, a number of efficiency tweaks may affect the execution time in the average case. For instance, the solutions to most frequent queries may be cached, or smart strategies may be employed to affect the way the solver explores a solution space. However, caching costs space, whereas any *rigid* resolution strategy may result efficient on some sorts of queries, while still being slow on some others.

In all those cases, sub-symbolic sub-systems capable to learn from experience can bring about huge benefits. There, a sub-symbolic helper may be trained to predict the outcomes of most frequent queries, thus speeding up queries with constant space requirements. Furthermore, an online learning procedure may be injected into the solver, making it adapt the resolution strategy to the query at hand, on the basis of the experience accumulated via previous queries.

This task may be particularly relevant when considering real-time agents working with complex knowledge bases. Focusing again on kinships, when the number of family members is huge and relations between family members are complex – e.g., fourth grade cousins –, query resolution may suffer from delays hindering agents ability to make real-time decision and perform real-time tasks. Therefore, it may be interesting to use techniques that aim at speeding up the resolution of queries over such huge theories. Sub-symbolical approaches may ease this task, by compressing theory knowledge to simple and easy-to-handle embeddings.

3.2. Graphs as Bridges

Here we discuss the role of GNN in addressing the relevant logic tasks from Section 3.1. In particular, we show how all such tasks can be mapped onto known graph-related problems which can be addressed via GNN. In other words, we comment the upper part of Figure 2.

Knowledge filling → Link prediction. The knowledge filling task usually requires semantic knowledge to be taken into consideration. Therefore, to map the knowledge filling task to an equivalent problem over graphs we should consider preserving the semantic information of the theory. We can then assume to map entities of a theory to vertices of a graph. Rules and relations can then be represented as vectorised arcs existing between the graph vertices. Each position of an arc vector represents a specific relation, preserving the original semantic of the theory. In this scenario, the task of predicting possible missing relations or rules is mapped to the problem of identifying which arcs are missing from the graph.

Knowledge Inclusion → Graph matching. In the same way as for the knowledge filling task, knowledge inclusion requires semantic knowledge of the theory to be taken into account. Therefore, we require the mapping between logic and graphs to preserve the theory semantic. Moreover, knowledge inclusion requires a comparison between two or more theories: entities and relations from a theory should be compared to their counterparts of the other theory and matched upon need.

As done for knowledge filling, let us assume entities to be represented as vertices, and rules or relations as vectorised arcs. The mapping produces as many graphs as the theories available for the inclusion task. Therefore, from a graph perspective, knowledge inclusion is mapped to a graph matching problem. Indeed, the two or more graphs corresponding to their theory counterparts should be matched for some portion of them.

The matching between graphs is still an open research problem, as it is computationally very expensive, but is easier to tackle than rules and entities matching. This holds in particular whenever entities do not match exactly, or, rules share analogous semantics but are defined in different forms—e.g., parent and mother.

Program equivalence \rightarrow **Graph compression.** Given a specific theory, program equivalence aims at obtaining a simpler – smaller – theory that preserve the same expressiveness. Depending on the considered approach the mapping between logic and graph level may bear different requirements. In its simplest form program equivalence requires to simply remove unnecessary relations and rules of a theory to compress it. This approach does not require explicitly the semantic level to be considered while processing the theory. More interestingly, program equivalence may also require to map set of rules and relations to a single (or a smaller set of) rules(s). This increased complexity introduces the need for semantic to be taken into account and to be preserved in the mapping from logic to graphs. If we consider the same mapping of previous examples, program equivalence can be linked to the graph compression problem. Indeed, obtaining a smaller set of equivalent rules and entities can be done removing or merging together arcs and vertices of the graph theory counterpart.

Query resolution speed-up \rightarrow **Graph classification.** Query resolution speed-up aims at obtaining faster execution of given queries over a logic theory. It may be helpful for query resolution to maintain the semantic information embedded in the theory. Therefore, the mapping between logic and graphs may benefit from the preservation of semantic information, and generally speaking vastly depends on the requirements of the desired speed-up. Differently from previous tasks, for query resolution speed-up we consider obtaining graphs for queries to be solved—rather than a single graph for the whole theory. The graph representing a query is matched with the query resolution, considered as the graph label. Following this mapping, the query resolution speed-up is mapped to a graph classification problem, where the label of a graph should be predicted. Any approach can then be leveraged to classify graphs—i.e. obtain query solutions. This approach may not be significant for simple queries applied to small knowledge basis. However, it may result in great speed-up when complex knowledge bases and queries are considered. Indeed, GNN scalability over large graphs is mostly not an issue, resulting in quick graph classification.

3.3. The Framework Perspective

Here we summarise the general framework to process logic theories sub-symbolically, via GNN. In a nutshell, the whole framework consists in *transforming* the problem into the graph domain and let a GNN to do the job, then possibly *transform* back the problem into the logic domain. The same framework is depicted in the lower part of Figure 2.

Let us assume the overall goal of the whole processing, at a logic level, is to perform a task T —say, knowledge filling or inclusion. Let us also assume that an adequate mapping exists for T towards the graph level, such that T' is graph-related task corresponding to T , at the graph level. Under such assumptions, the framework involves the following steps:

1. a logic theory must be encoded into a graph using a suitable graph *encoding schema*;
2. a GNN must be designed and trained to perform T' , choosing
 - the functions Ξ^t , Θ , and Θ^t for the GNN *convolutor* block,
 - the structures of the GNN *aggregator* and *predictor* blocks;

3. optionally, the output of the GNN shall be decoded back into a logic theory using a suitable graph *decoding schema*.

The emphasised words above represent choice points for the designer. While the possibilities are manifold, it is worth pointing out that each choice affects the others. For instance, while the encoding schema should be chosen by taking the nature of the logic clauses into account, the architecture of the predictor block, as well the choice of functions \oplus and \ominus^t , should be tailored on the task T' —and in particular on its nature under the learning perspective (e.g. whether T' is classification, regression, or clustering task).

Whether it is needed to perform step 3 (decoding) or not, is another source of constraints. There may be tasks – such query resolution speed-up, corresponding to graph classification – for which the outcome of T' is a Boolean datum, which needs not a decoding step. Conversely, other tasks may require the outcome of T' to be transformed back into the logic domain—cf. knowledge inclusion via link prediction. When this is the case, it is of paramount importance to choose an encoding schema which is *invertible*. This implies the encoding and decoding schemas are deeply entangled in the general case.

Summarising, symbolic processing may greatly benefit from the exploitation of sub-symbolic, GNN-based approaches. However, when this is the case, the overall data-processing framework must be carefully designed, as it involved may inter-dependent design choices.

4. Case Study

In this section we describe a case study that puts the theoretical framework introduced in Section 3 to test. We consider the knowledge filling task as the subject of this case study, as we believe it to be a nice introductory example to the world of logic manipulation using GNN. We proceed to set up our study case over a controlled environment, considering the knowledge basis representing kinship relations—i.e. family tree. We “mutilate” the knowledge base – meaning that we throw away a random part of the knowledge base – and use the remaining part to reconstruct the information removed by using the proposed framework.

4.1. Logic to Graph

As already mentioned, measuring the effectiveness of our framework requires to “mutilate” an otherwise exhaustive knowledge base. Theory mutilation can be then attained both at a logical level and at the graph level. In our experiments we mutilate the theory at graph level, to avoid multiple translations between the two levels. We now introduce the mapping function between logic level and graph level used in our experiment.

Translation to Graph Let \mathcal{C} be the set of all ground Horn clauses of the form $h \leftarrow b_1 \wedge \dots \wedge b_m$ s.t. all b_i as well as h are predicates of arity non-greater than 2, and all arguments of all predicates are constant in \mathcal{H} . We consider $\tau \in \mathcal{C}^*$ to be a ground theory containing N clauses and representing a family tree. We then define the *properties* of the theory τ to be the set of all the unary predicates mentioned in all clauses of τ . The set of properties is considered to be ordered through an index k , allowing to obtain a property calling it with the corresponding

```

sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.
united(X, Y) :- parent(X, Z), parent(Y, Z), X \= Y.
grandparent(C,D) :- parent(C,E), parent(E,D).
aunt(X, Y) :- female(X), sibling(X, Z), parent(Z, Y).
uncle(X, Y) :- male(X), sibling(X, Z), parent(Z, Y).

male(matt).
male(theo).
male(joseph).

female(susy).
female(lisa).
female(jane).

parent(matt, joseph).
parent(susy, joseph).
parent(jane, matt).

sibling(susy, theo).
sibling(theo, susy).
sibling(matt, lisa).
sibling(lisa, matt).

```

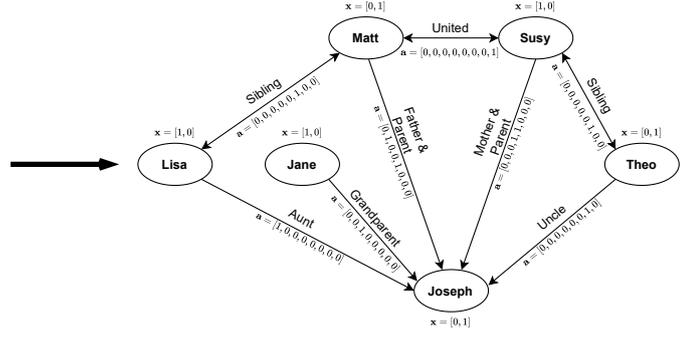


Figure 3: Example of the mapping function used in this case study. vertex attributes (x_v) represent unary predicates – i.e. properties – while arc attributes ($a_{v,w}$) represent binary predicates—i.e. relations.

index. We then define the *relations* of the theory τ to be the set of all the binary predicates mentioned in all clauses of τ . The set of relations is also considered to be ordered through an index l , allowing to obtain a relation calling it with the corresponding index.

To map the theory τ to its graph counterpart G_{fill} , we first consider all the entities mentioned in all clauses of τ and associate a vertex to each entity. Therefore, obtaining a graph G_{fill} with n vertices. Vertex features are then built as vectors $x_v \in \mathbb{R}^d$, where d represents the size of the set of *properties* of τ . Vector x_v has value at position $k \in \{1, \dots, d\}$ equal to 1 if property k holds true for entity v and 0 otherwise. The obtained vertex feature vector is thus a one-hot encoded vector representing the properties that characterise the entity. Similarly to vertices, arc features are built as vectors $a_{v,w} \in \mathbb{R}^c$, where c represents the size of the set of *relations* of τ . Vector $a_{v,w}$ has value at position $l \in \{1, \dots, c\}$ equal to 1 if relation l between entities v and w holds true and 0 otherwise. The obtained arc feature vector is thus a one-hot encoded vector representing the relations satisfied for couples of vertices. Figure 3 exemplifies the mapping described above for a small family tree. In our experiment, predicates are the same of the figure, along with some added unary predicates – e.g. *has_siblings*, *is_parent*, *is_grandparent*, etc. –, which are used to give more information concerning single entities.

Once G_{fill} is obtained, we proceed to mutilate the theory. Mutilation is attained removing some arcs – i.e. relations – between vertices of the graph. We call the graph obtained through this procedure G and the set of removed arcs A_{test} .

The proposed mapping allows constructing uniquely a graph from a grounded knowledge bases and is evidently bijective, as it is possible to reconstruct entities, properties and relations from x_v and $a_{v,w}$. However, it still presents some issues, as it requires groundisation of the knowledge bases and it can handle at most binary facts. The former can be considered a mild requirement as it is commonly considered for manipulation of symbolic knowledge. The latter instead has to be attributed to the nature of state-of-the-art GNN. Predicates of arity greater than 2 would require arcs linking more than two vertices at the time. Graphs having such links are called hypergraphs. These peculiar graphs are still an exception in the world of graph manipulation. There have been proposed very few solutions to handle these graphs [15], presenting strong drawbacks like the absence of arc features.

4.2. Graph Manipulation

Given the mutilated theory represented by the graph G , the task is to train a GNN model capable of mining the missing arcs. The GNN model is required to identify not only the existence of a missing arc between two vertices, but also to classify the arc into its class—i.e. which relation(s) the arc is representing. Due to graphs nature, link prediction can be tackled either considering G as a unique entity – i.e. *plain approach* – or as a pool of subgraphs—i.e. *subsampling approach*.

Plain approach. We consider the graph G as a whole and predict one solution for each couple of vertices. Indeed, each vertex can be involved in a relation with any other vertex of the graph. Here, we consider the set of arcs belonging to G as positive examples, called A . We then sample a set of negative examples \bar{A} as all the arcs that are not in A , nor in A_{test} . Given in input the graph G , a GNN model is then trained over A and \bar{A} to output two predictions:

- A binary matrix $Y_e \in \mathbb{R}^{n \times n}$. Where n is the number of entities in the graph. The value for position $\{v, w\}$ is 1 if the GNN predicts that an arc should exist between vertex v and vertex w and 0 otherwise.
- A binary tensor $Y_t \in \mathbb{R}^{n \times n \times c}$. Where n is the number of entities in the graph and c is the number of available facts (kinship relations). The c -dimensional vector at position $\{v, w\}$ corresponds to the one-hot encode of the arc type that the GNN predicts between vertices v and w .

The GNN model is composed of two graph convolutional layers that extract relevant information from G and produce a graph embedding G' . Given the need to predict a solution for each couple of vertices, we define as aggregation function the concatenation of vertices in the graph. The function is iterated over each couple of vertices v, w producing a vector $\bar{x}_{v,w} = \mathbf{x}'_v \parallel \mathbf{x}'_w$ that represents the embedding for a possible arc between vertices v and w . Two parallel fully connected layers are then used as predictors to predict the existence – i.e. Y_e – of arc v, w and its type – i.e. Y_t – from $\bar{x}_{v,w}$. During training, cross-entropy loss \mathcal{L}_e is computed using Y_e , while binary cross-entropy over class types is \mathcal{L}_t is computed using Y_t [16]. The overall loss is then obtained through weighted summation of the two and used to optimize the GNN parameters.

$$\mathcal{L} = \mathcal{L}_e + \gamma \mathcal{L}_t \quad (2)$$

where γ is an hyperparameter balancing the importance of predicting arc existence or its type.

Subsampling approach. Similarly to [17], it is possible to consider the graph G as a pool of subgraphs each of which is used to predict if one arc exists. Here, one subgraph G_{sub} is obtained for each arc in A . For each arc $a_{v,w}$, the subgraph is obtained by keeping vertices v and w , as well as their neighbours $N(v)$ and $N(w)$. The same sampling procedure is repeated for a set of negative examples \bar{A} . Therefore, following this approach we obtain a set $\mathcal{G} = \{G_{sub}^1, \dots, G_{sub}^\nu\}$ of ν graphs, each focused on an arc $a_{v,w}$.

A GNN model is then trained over A and \bar{A} , receiving in input a graph G_{sub} at the time, to output two predictions:

- A binary value $Y_e \in \{0, 1\}$. The value is 1 if the GNN predicts that the arc $a_{v,w}$ should exist and 0 otherwise.
- A binary vector $Y_t \in \mathcal{R}^d$. Where c is the number of available facts. The vector corresponds to the one-hot encode of the arc type that the GNN predicts for arc $a_{v,w}$.

The GNN model is similar to the one of plain approach. It is composed of two graph convolutional layers that extract relevant information from G_{sub} and produce a graph embedding G'_{sub} . Given the need to predict a single solution for each graph, we define as aggregation function the global averaging pooling of vertices in the graph. Global average pooling produces in output a k -dimensional vector $\bar{\mathbf{x}}$, averaging vertex features of all vertices in the graph G'_{sub} , $\bar{\mathbf{x}} = \frac{1}{N} \sum_{v=1}^N \mathbf{x}'_v$. Two parallel fully connected layers are then used as predictors to predict the existence – i.e. Y_e – of arc v, w and its type – i.e. Y_t – from $\bar{\mathbf{x}}$. GNN of both approaches are built using *PyTorch Geometric* [18] and *Deep Graph Library* [19]. Finally, loss computation remains the same of plain approach.

Considerations. It must be stressed that link prediction over graph is usually considered to be a binary prediction problem only. Indeed, state-of-the-art approaches focus only on the output Y_e . This is due to the nature of common link prediction applications – e.g. chemistry, social networking – where arcs between vertices belong mainly to one category only. As a consequence, the link prediction problem we face is more complex, as it represents the mixture of binary classification and multi-label classification—as there may exist multiple relations linking two entities. Tackling multi-label classification is not straightforward due to class overlapping and scalability issues [20].

It would be desirable to have relations semantically distant from each other to aid GNN in the multi-label classification task. Indeed, classes characterised by similar semantics are less separable and are probably subject to higher misclassification. Moreover, the scalability issue of multi-label classification hinders the performance of GNN when considering a high number of relation types. Although they could be overcome, these issues must be taken into account during the evaluation of the proposed experiment.

4.3. Results

During GNN training procedure we split either G (plain approach) or \mathcal{G} (subsampling approach) between training set and validation set. The former is used for backpropagation, while the latter is used to check GNN performance and save the best model. The best model obtained from training is then applied over A_{test} to test the final performance of the GNN.

We measure model performance over both predictions tasks—i.e. over Y_e and Y_t . We measure how the model behaves for the arc existence prediction task using the Area Under the Curve (AUC) [21, 22], Average Precision (AP) and accuracy metrics.

Evaluation of multi-label classification is not straightforward, as it introduces the notion of partially correct prediction—i.e. those predictions where a subset of the labels are identified correctly, but not all of them. To measure how the model behaves in the arc type prediction task we leverage the well known F_1 score, the Exact Match Ratio (EMR), and the Per Example Accuracy (PEA). EMR ignores partially correct predictions, considering them as incorrect and

Approach	Edge Existence			Edge Class		
	AUC	AP	Accuracy	EMR	F ₁	PEA
Plain	0.786	0.786	0.786	0.143	0.148	0.768
Subsampling	0.908	0.892	0.906	0.691	0.386	0.911

Table 1
Performance of the two different approaches over A_{test} .

computes the score as the ratio between correct predictions and total predictions. On the other hand, PEA computes the accuracy of the single sample i for each i in A_{test} and then average them together to obtain an overall accuracy metric.

Table 1 shows the performance of the two approaches on the link prediction task over A_{test} . Obtained results demonstrate the effectiveness of the proposed approach. The prediction of arc existence is successful with acceptable level of performance for the plain approach, while being highly successful when graph subsampling is applied. On the other hand, for the arc type prediction task we can notice the clear superiority of the subsampling approach.

Subsampling approach superiority can probably be attributed to the setup of the learning task. For the plain approach, the same input G is used to predict arcs belonging to training, validation and test, increasing the tendency of overfitting. On the other hand, subsampling approach allows to assign a different input $G \in \mathcal{G}$ to each arc to predict. These inputs are different between training, validation and test, therefore allowing the model to train more easily, avoiding possible overfitting issues.

Given the number of possible arcs types (8) and their possible overlapping – e.g. *parent* and *father* –, the performance obtained by the subsampling approach are very satisfactory. The proposed model is capable to completely match an arc to its label – i.e. exact match over all 8 classes that define the arc – nearly 70% of the times, while the single arc types are predicted correctly more than 90% of the times. These results show the effectiveness of both the GNN model and the proposed theoretical framework of Section 3.

Figure 4 shows the three possible outcomes of arc predictions. An arc prediction may be completely correct—e.g. *Gabriel* \rightarrow *Julia*. There may also exist partially correct predictions—e.g. *Gabriel* \rightarrow *Albert*. Finally, *Lisa* \rightarrow *Gabriel* shows that there may exist arcs wrongly predicted.

Figure 4 also highlights a peculiar property of the model. The *Lisa* \rightarrow *Gabriel* prediction considered as wrong is actually a *daughter* relation. This prediction is considered as wrong since neither *daughter* nor *son* are defined in the original theory τ . Indeed, to solve the knowledge filling task we rely only on the predicates already defined. However, the ability of the model of predicting this relation is sign of the GNN ability to understand the semantic of τ . This GNN capability is encouraging, as it may help in tasks such as the discovery of new predicates.

Role of overlapping classes. As shown by Figure 3, the results obtained for Table 1 are influenced by class overlapping. Indeed, some arc types considered are semantically similar—e.g. *mother* and *parent*. Therefore, their vectorial representation produces label overlapping, which may negatively affect GNN performance.

To study the effect of semantically-similar classes on our approach, we consider a new theory

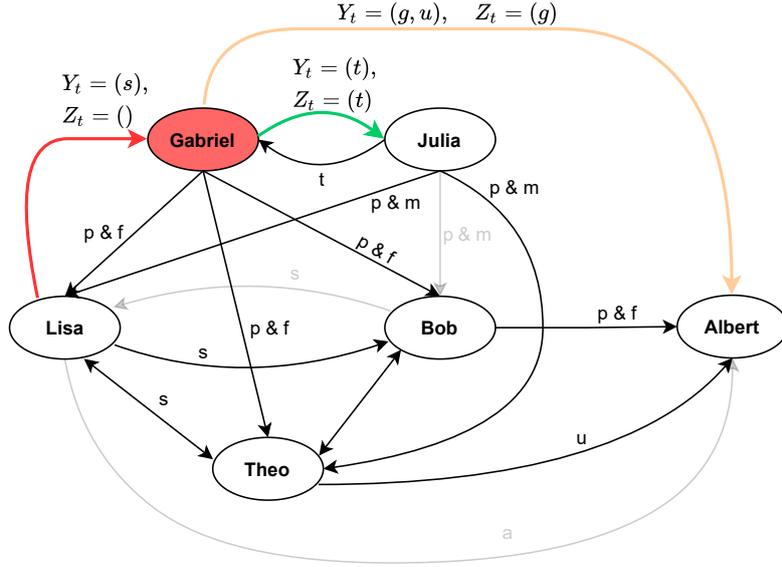


Figure 4: An example of three arc predictions involving entity *Gabriel*. Vertices represent entities and arcs represent relations, which can be aunt (a), father (f), grandparent (g), mother (m), parent (p), sibling (s), together (t), uncle (u). Black arcs and relations identify arcs in G , while gray arcs are arcs belonging to A_{test} and used for testing. Coloured arcs are the ones belonging to A_{test} and involving *Gabriel*. For these arcs Y_t and Z_t represent model prediction(s) and label(s) respectively. Green colour means the arc is predicted correctly, orange identifies partially correct arcs, while red pinpoint arcs wrongly predicted.

Approach	Edge Existence			Edge Class		
	AUC	AP	Accuracy	EMR	F ₁	PEA
<i>Plain</i>	0.857	0.857	0.857	0.286	0.214	0.738
<i>Subsampling</i>	0.949	0.961	0.938	0.670	0.340	0.935

Table 2

Performance of the two different approaches over A_{test} when overlapping arc types – e.g. *father* and *parent* – are not considered.

τ' identical to τ – i.e., representing the same family tree –, but formulated without semantically-similar predicates. Practically speaking this requires to remove the notion of *father* and *mother* from τ and redefine kinships using only the notion of *parent*.

We apply the same approaches of Section 4.2 to the new theory τ' , and measure their performance. Table 2 shows the performance of the two approaches on the link prediction task over A_{test} when overlapping arc types are not considered. As expected, the superiority of the subsampling approach is unaffected. Instead, it is interesting to point out the effects of overlapping relations on the arc prediction task. The removal of overlapping arc types seems to affect positively *Edge Existence* prediction. Arc existence prediction compresses the knowledge of arc type to its mere existence. Therefore, overlapping relations might confuse the model in this scenario.

On the other hand, metrics related to the task of classifying arc type – i.e. *Edge Class* – seem to be untouched. This is a positive indication, as it hints at the fact that GNN can handle overlapping relations between entities. However, further study of this behaviour for the proposed model is required.

5. Conclusions

In this paper we propose a theoretical framework leveraging translation techniques between logic theories and graphs so as to tackle relevant logical problems. We identify the relevant logical problems and describe how they can benefit from the use of sub-symbolical approaches. Along with our framework, we identify possible mappings between logic and graphs, stressing their properties and defining how these should be identified by users. We then consider the filling of a fragmented theory as a study case and apply the proposed framework to this task. Obtained results show the goodness of our approach, introducing the possibility to leverage GNN to identify missing latent relations between entities of a theory. Finally, a brief study on GNN behaviour for overlapping classes in link prediction problem is presented, showing hints of GNN reliability.

Future works should focus on applying our framework for tackling relevant logical tasks in the CL world. Moreover, we consider relevant for future works to focus on some limitations of GNN, emerged from the work on our framework. Many logical tasks require mapping of predicates having arity greater than two to hypergraphs. Furthermore, some logical tasks require considering many relations, mapped to highly dimensional arc vectors. State-of-the-art GNN still suffer these requirements, especially when combined. We believe that future research in GNN should propose models capable of working on complex highly-dimensional hypergraphs. Finally, given the centrality role of arc relations in our framework, it would be desirable to have GNN models more focused on arc attributes relevance. Indeed, information is commonly updated only at vertices level, while logical tasks, and GNN performance more in general, would benefit from updating arc information as well. Few works so far have tackled this issue [23, 24], characterised by strong requirements and poor generalisability. Therefore, it exists research potential leveraging arc information more efficiently in GNN.

Acknowledgments

This paper has been partially supported by (i) the H2020 project “StairwAI” (G.A. 101017142), and (ii) the CHIST-ERA IV project “EXPECTATION” (G.A. CHIST-ERA-19-XAI-005).

References

- [1] A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, *Information Fusion* 58 (2020) 82–115. doi:10.1016/j.inffus.2019.12.012.

- [2] R. Calegari, G. Ciatto, A. Omicini, On the integration of symbolic and sub-symbolic techniques for XAI: A survey, *Intelligenza Artificiale* 14 (2020) 7–32. doi:10.3233/IA-190036.
- [3] F. J. Kurfess, Integrating symbol-oriented and sub-symbolic reasoning methods into hybrid systems, in: *From Synapses to Rules*, Springer, 2002, pp. 275–292.
- [4] M. L. Minsky, Logical versus analogical or symbolic versus connectionist or neat versus scruffy, *AI Magazine* 12 (1991) 34. doi:10.1609/aimag.v12i2.894.
- [5] L. Serafini, A. S. d’Avila Garcez, Logic tensor networks: Deep learning and logical reasoning from data and knowledge, in: T. R. Besold, L. C. Lamb, L. Serafini, W. Tabor (Eds.), *NESY 2016 – Neural-Symbolic Learning and Reasoning*, volume 1768 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2016. URL: http://ceur-ws.org/Vol-1768/NESY16_paper3.pdf, proceedings of the 11th International Workshop on Neural-Symbolic Learning and Reasoning (NeSy’16) co-located with the Joint Multi-Conference on Human-Level Artificial Intelligence (HLAI 2016).
- [6] I. Donadello, L. Serafini, A. S. d’Avila Garcez, Logic tensor networks for semantic image interpretation, in: C. Sierra (Ed.), *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, Melbourne, Australia, August 19-25, 2017, ijcai.org, 2017, pp. 1596–1602. URL: <https://doi.org/10.24963/ijcai.2017/221>. doi:10.24963/ijcai.2017/221.
- [7] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A comprehensive survey on graph neural networks, *IEEE Transactions on Neural Networks Learning Systems* 32 (2021) 4–24. doi:10.1109/TNNLS.2020.2978386.
- [8] Z. Shui, G. Karypis, Heterogeneous molecular graph neural networks for predicting molecule properties, in: C. Plant, H. Wang, A. Cuzzocrea, C. Zaniolo, X. Wu (Eds.), *20th IEEE International Conference on Data Mining, ICDM 2020*, Sorrento, Italy, November 17-20, 2020, IEEE, 2020, pp. 492–500. doi:10.1109/ICDM50108.2020.00058.
- [9] H. Wang, J. Leskovec, Unifying graph convolutional neural networks and label propagation, *CoRR abs/2002.06755* (2020). arXiv:2002.06755.
- [10] W. Fan, Y. Ma, Q. Li, Y. He, Y. E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, in: L. Liu, R. W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, L. Zia (Eds.), *The World Wide Web Conference, WWW 2019*, San Francisco, CA, USA, May 13-17, 2019, ACM, 2019, pp. 417–426. doi:10.1145/3308558.3313488.
- [11] R. Hecht-Nielsen, Theory of the backpropagation neural network, *Neural Networks* 1 (1988) 445–448. doi:10.1016/0893-6080(88)90469-8.
- [12] L. C. Paulson, Computational logic: its origins and applications, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474 (2018) 20170872. doi:10.1098/rspa.2017.0872.
- [13] T. Eiter, M. Fink, H. Tompits, S. Woltran, Simplifying logic programs under uniform and strong equivalence, in: V. Lifschitz, I. Niemelä (Eds.), *Logic Programming and Nonmonotonic Reasoning, 7th International Conference, LPNMR 2004*, Fort Lauderdale, FL, USA, January 6-8, 2004, *Proceedings*, volume 2923 of *Lecture Notes in Computer Science*, Springer, 2004, pp. 87–99. doi:10.1007/978-3-540-24609-1_10.
- [14] J. Ji, H. Wan, Z. Huo, Z. Yuan, Simplifying A logic program using its consequences, in: Q. Yang, M. J. Wooldridge (Eds.), *24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, AAAI Press, Buenos Aires, Argentina, 2015, pp. 3069–3075. URL:

<http://ijcai.org/Abstract/15/433>.

- [15] S. Bai, F. Zhang, P. H. S. Torr, Hypergraph convolution and hypergraph attention, *Pattern Recognition* 110 (2021) 107637. doi:10.1016/j.patcog.2020.107637.
- [16] J. E. Shore, R. W. Johnson, Properties of cross-entropy minimization, *IEEE Transactions on Information Theory* 27 (1981) 472–482. doi:10.1109/TIT.1981.1056373.
- [17] M. Zhang, Y. Chen, Link prediction based on graph neural networks, in: S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018*, pp. 5171–5181. URL: <https://proceedings.neurips.cc/paper/2018/hash/53f0d7c537d99b3824f0f99d62ea2428-Abstract.html>.
- [18] M. Fey, J. E. Lenssen, Fast graph representation learning with pytorch geometric, *CoRR abs/1903.02428* (2019). arXiv:1903.02428.
- [19] M. Wang, L. Yu, D. Zheng, Q. Gan, Y. Gai, Z. Ye, M. Li, J. Zhou, Q. Huang, C. Ma, Z. Huang, Q. Guo, H. Zhang, H. Lin, J. Zhao, J. Li, A. J. Smola, Z. Zhang, Deep graph library: Towards efficient and scalable deep learning on graphs, *CoRR abs/1909.01315* (2019). arXiv:1909.01315.
- [20] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, W. Xu, CNN-RNN: A unified framework for multi-label image classification, in: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, pp. 2285–2294. doi:10.1109/CVPR.2016.251.
- [21] J. A. Hanley, B. J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, *Radiology* 143 (1982) 29–36.
- [22] A. Clauset, C. Moore, M. E. J. Newman, Hierarchical structure and the prediction of missing links in networks, *Nature* 453 (2008) 98–101.
- [23] L. Gong, Q. Cheng, Exploiting edge features for graph neural networks, in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, Computer Vision Foundation / IEEE, 2019, pp. 9211–9219. doi:10.1109/CVPR.2019.00943.
- [24] X. Jiang, R. Zhu, P. Ji, S. Li, Co-embedding of nodes and edges with graph neural networks, *CoRR abs/2010.13242* (2020). arXiv:2010.13242.

Embedding a Neuro-Fuzzy Mode Choice Tool in Intelligent Agents

Maria Nadia Postorino¹, Giuseppe M. L. Sarné² and Mario Versaci³

¹Department DICAM, Alma Mater Studiorum University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy

²Department of Psychology, University of Milan Bicocca, Piazza dell'Ateneo Nuovo, 1, 20126 Milan, Italy

³Department DICEAM, University Mediterranea of Reggio Calabria, Loc. Feo di Vito, 89122 Reggio Calabria, Italy

Abstract

Increasing road traffic levels in urban areas require actions and policies to manage and control the number of road users. Travelers' choices of transport modes, particularly private cars, that generate the main share of road traffic levels, depend on many factors, which include both personal preferences and level-of-service variables. Understanding how travelers choose transport modes according to the above factors is an important challenge in order to adopt the most suitable policies and facilitate a sustainable mobility. In the literature, behavioral models have been mainly proposed in order to both estimate mode choice percentages and capture travel behaviors by suitable estimation of some parameters associated to the above factors. However, behavior is complex in itself and the mechanisms underlying user behavior might be difficult to be captured by traditional models. In this paper, a neuro-fuzzy approach is proposed to extract mode choice decision rules by evaluating different sets of rules and different membership functions of the neuro-fuzzy model. Particularly, to determine which inputs are the most relevant in such decision process, fuzzy curves and surfaces have been considered in order to take into account nonlinear effects. The neuro-fuzzy model proposed in this paper has been thought to be embedded in an agent-based methodological framework where user agents – representing travelers – make travel choices based on the rules learnt by means of the neuro-fuzzy system.

Keywords

Agent System, Fuzzy System, Mode choice, Neuro-Fuzzy Inference, Rule Learning

1. Introduction

Traffic flow conditions in urban road networks are the consequences of several user's choices – from the decision to own a private vehicle to the decision to use it for commuting or to move between origin/destination pairs, at different periods, along some paths. In the last decades, road vehicle traffic levels have been constantly increasing due to – among the others – better economic conditions, which has led to an increasing number of personal vehicles, and increasing number of urban inhabitants, which has implied a continuous growth of urban mobility.

WOA 2021: Workshop "From Objects to Agents", September 1–3, 2021, Bologna, Italy

✉ marianadia.postorino@unibo.it (M. N. Postorino); giuseppe.sarne@unimib.it (G. M. L. Sarné);

mario.versaci@unirc.it (M. Versaci)

🌐 <https://www.unibo.it/sitoweb/marianadia.postorino> (M. N. Postorino);

<https://www.unimib.it/giuseppe-maria-luigi-sarne> (G. M. L. Sarné);

https://www.unirc.it/scheda_persona.php?id=650 (M. Versaci)

🆔 0000-0002-6609-4818 (M. N. Postorino); 0000-0003-3753-6020 (G. M. L. Sarné); 0000-0003-3837-6671.

(M. Versaci)



© 2020 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

The consequences of increasing vehicular traffic levels are well-known, ranging from traffic jams, loss of time and economic resources – which affect the quality of life of citizens – to environmental effects both as a direct consequence of mobility (i.e., air pollution) and indirectly for the use of environmental resources from production to end-of-life of a vehicle [1, 2, 3, 4].

Different strategies have been adopted at different levels (e.g., local, national and supranational) to manage such effects, mainly by providing solutions based on transit modes and new technologies [5, 6, 7, 8]. However, suitable policies aimed at balancing private vehicular traffic against shared transportation modes require prediction of mode users' choices based on both level-of-service variables and personal preferences [9, 10]. In addition, user's path choices affect the way in which vehicular congestion spreads across the transportation network [11]. From one hand, the choice to use a private vehicle involves also mobility choices (e.g., having a guide license, owning a vehicle) [12], on the other hand, once the private vehicle has been chosen as modal alternative, users will get their trip destinations from given trip origins by following a sequence of road facilities, which identify paths on the transportation networks [13, 14, 15].

As well known, ultimately path choices will lead to traffic flow levels on road links [16, 17]. Then, the sequence transport mode–path choices has important consequences, both directly on the transportation network conditions (e.g., travel times, pollution, monetary costs) and indirectly on the territorial system (e.g., environmental impacts, life quality).

Analysis and prediction of users' choices – mainly mode choices - are considered fundamental both for the knowledge of travel demand on the available transport modes and for deciding how to vary the level-of-service variables of the transportation supply in order to achieve sustainable mobility and better quality of life [18].

Every day users perform several choices about their trips (e.g., destination, travel mode and path), which depend on the combination of policy measures (e.g., traffic restrictions, access fees to Limited Traffic Zones, parking fares, cost of public transport), external issues (e.g., fuel price) and new forms of mobility like car-sharing, car-pooling, and, as expected in the next future, shared connected and autonomous vehicles [19, 20]. To realize their trips most users still prefer to use private mobility rather than transit opportunities. In other words, individual drivers are reluctant to change their habits regardless of the policies adopted and the additional costs involved in using private mobility. Therefore, it is clear that one key issue is to understand user's travel behavior more in-depth as well as the factors that influence it.

Understanding users' choice behavior is becoming a pressing need and many models and approaches have been used to this aim, from random utility models [21, 22, 23] to stated preferences (SP) [24] techniques and artificial intelligent agents [25]. Random utility models associate a utility value to each available alternative depending on some attributes that characterize the alternative itself. Then, it is assumed that users choose the alternative having the maximum perceived utility, which is in fact a stochastic variable whose distribution function leads to several discrete choice models (for example, Logit, Nested Logit and Probit models [23, 26, 27], the derived Dogit Logit [28] and Logit Box-Cox transformation [29, 30], as well as the the Cross-Nested Logit or the Generalized Nested Logit models [31, 32]). Such models are often based on compensatory approaches [33] where negative attributes (e.g., times or costs) balance positive attributes (e.g., reliability or comfort), both suitably weighted by parameters that result from calibration procedures exploiting user's choice data. A promising alternative to random utility approaches is represented by neuro-fuzzy models [34, 35, 36], which represent

a class of adaptive networks that combine the ability to generate a fuzzy inference system (FIS) with a linear relationship in input-output data given by the neural network. Neuro-fuzzy models are known for their ability to capture the vagueness and ambiguity inherent in human decision-making processes.

Additionally, in recent years there has been an increasing use of intelligent software agents (hereafter simply agents) in the transportation domain, to cover manifold aspects and provide effective and efficient solutions for Intelligent Transportation Systems (ITS) [37, 38, 39] as, for instance, to obtain trip choice probabilities [40] or to estimate perceived travel time [41] among the others. In particular, agents are designed with learning, cooperative and adaptive behaviors capabilities [25, 42, 43], often based on Artificial Intelligence (AI) techniques [44], which make them suitable to simulate a great variety of complex human behaviors and agent-to-agent interactions at different levels of detail [45, 46, 47] and abstraction [48]. Neuro-fuzzy systems can be embedded into software agents that need to make decisions by simulating human behavior.

In this perspective, this paper proposes to adopt a neuro-fuzzy component able to both identify the main key factors and model user's mode choice behavior [49, 50, 51, 52], which is expected to be used as input for modular agents simulating travelers' behaviors. More in detail, the paper proposes a neuro-fuzzy network approach to identify behavioral rules that will be used by agents in a cascading structure. Although the article focuses only on the neuro-fuzzy model and its results to understand and model travel mode choice behavior, the modular structure of the agent will also be described in order to provide the methodological framework in which the proposal was developed. Some simulations have been carried out to estimate the performance of the proposed neuro-fuzzy component. The obtained results are satisfactory and the designed neuro-fuzzy component to simulate the transportation modal choice has the undeniable advantage of making explicit in a clear and unequivocal way the users' behavioral rules, which otherwise would have remained embedded in the parameters of the traditional models. Explicitly understanding behavioral rules that lead users to some travel choices will make it possible, on the one hand, to meet better user's expectations in terms of transport services, and, on the other, to meet the requirements of sustainable mobility.

The rest of the paper is organized as follows. The next Section introduces the agent-based approach in a modular framework perspective. Section 3 describes the neuro-fuzzy modeling while Section 4 gives an overview on the adopted methodology. Finally, in Section 5 the neuro-fuzzy based module is validated and in Section 6 some conclusions are drawn.

2. The Agent-based Approach: Modular Framework

This Section provides the agent-based methodological framework in which the neuro-fuzzy model is embedded. The basic idea underlying this simulator is that traffic flows on several specific transportation mode (e.g., private, public and pedestrian) might be simulated in an integrated way starting from mode choices made by travelers based on both socio-economic and transport supply features, while usually they are simulated in a separated way.

As introduced in Section 1, mode choices affect urban traffic levels in terms of next path choices, which are linked to the features of the transportation supply mainly in terms of level

of service variables. As mode choices generally depend on user socio-economic characteristics other than on mode supply features, here agents have been associated with users rather than vehicles. Starting from this perspective, the decision-making process – described in the following sections – autonomously performed by the users (i.e., user agents) to choose a specific transportation mode to accomplish their trips has been modelled mainly based on these two groups of variables - i.e., socio-economic and transport supply variables. The agent-based approach simulates the travel decision process starting from the first step, i.e. learning behavioral rules for being able to make mode choices. It overcomes some limits of a previous simulator developed by the Authors, which even though has a high versatility (it was used also to simulate *Urban Air Mobility* (UAM) for point-to-point connections [53, 54]), from the other hand does not allow further progresses like investigating on users' behavior.

The hypothesis of “rationality” has been assumed, similarly to random utility models, which implies that each user agent is able to make a choice based on the advantages and disadvantages of each available alternative. Figure 1 represents the block-scheme of the user agent. More in detail, the initial information, which refers to behavioral rules learnt by means of a neuro-fuzzy model, is provided by the (*Input Data*) block, whose task is to collect information (e.g., trip origin and destination, time, comfort, user's preferences, etc.) and sent to the *Mode Choice Module* block. This block simulates the user's selection of the transportation mode (here private, public and pedestrian modes have been considered). For each available transportation mode, a dedicated module (e.g., the agent module *Private Driving Module* for the private car mode) will interact with the corresponding transportation network (i.e., the *Private Transportation Network*, set as exogenous system where other user agents that are active on that network at a given time interact among them). Finally, the *Feedback Module* will release a feedback about information concerning the travel features on the given network experimented during the journey between the origin/destination pair. Such feedback will be used by the user agent to modify, if it is the case, his/her mode choices based on both the experiences and the set of rules learnt at the beginning and provided by the *Input Data* block [55].

Currently, we are developing the *Mode Choice Module* that equips the user agent and will realize the first task carried out by the user agent. In the adopted cascading structure, this module influences all the other simulations of the transportation network and, therefore, it represents the most critical component to simulate urban mobility by using an agent approach. This component will be explained in detail in the following.

3. The Neuro-based Fuzzy Inference System

In mathematics, a set is defined as a collection of objects independently by their number. In 1965, Lotfi Zadeh [56] proposed the idea of fuzzy set as a set to which objects can belong with different degrees of membership. The confidence that the element belongs to the fuzzy set is represented by a membership function whose values can range from 1 (absolutely true) to 0 (absolutely false) and that can assume different shapes (e.g., triangular, trapezoidal, gaussian, sigmoidal, etc.). Here, we will apply the fuzzy theory to sets of descriptive words, where the degree of membership identifies the confidence in the descriptor (i.e., its weight), often set by the analysts.

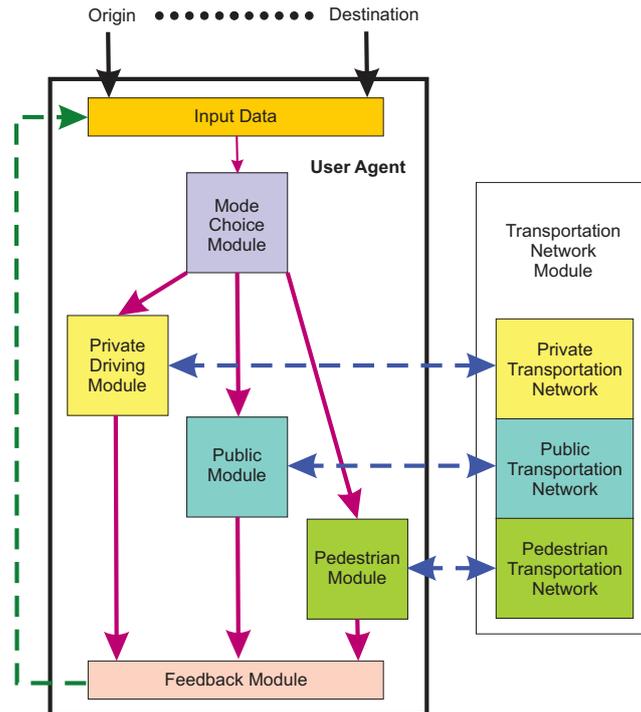


Figure 1: User Agent - Block scheme.

More in detail, a Fuzzy Inference System (FIS) is a set of fuzzy membership functions and rules to generate an output using fuzzy set theory, thus mapping an input space into an output space. The structure of a fuzzy inference system [57] includes three conceptual components:

1. a selection of fuzzy rules;
2. a catalog of membership functions exploited in the fuzzy rules;
3. a reasoning method, which executes the inference procedure (i.e., the fuzzy reasoning introduced earlier) by using the rules and a given condition to produce a logical output or conclusion.

In addition, the basic FIS can receive both fuzzy and crisp¹ inputs, returning (usually) fuzzy sets as output, but sometimes it is necessary that a crisp output (representative of the fuzzy set) is returned. In particular, when a FIS receives and returns crisp data, it makes a nonlinear mapping from the input space to the output space.

This mapping is made explicit through a set of fuzzy *if-then* rules. Each rule is referred to a local behavior of the mapping, where the antecedent of the rule delimits a fuzzy area of the input space, while the consequent refers to the associated outputs [58]. Various types of inference are described in the literature [59], but due to its flexibility in terms of writing, Sugeno

¹Crisp inputs are assimilated to fuzzy singletons with degree of membership 0 everywhere except at specific points where degrees of membership are 1.

inference [60, 61] will be adopted (see below) since it provides a systematic approach to obtain fuzzy rules for a given set of input-output data. More specifically, a generic fuzzy Sugeno rule has the form:

if x is A and y is B then z

where the antecedent consists of the fuzzy sets A and B, while the consequent is given by $z = f(x, y)$, where $f(\cdot)$ is a crisp function of x and y.

Generally, the process to make a FIS, usually called fuzzy modeling, consists of four steps:

1. *Fuzzification* - This process associates the degree of membership of an input data to a fuzzy set by means of special functions, called membership functions, defined on the range of possible values in the domain [0;1];
2. *Fuzzy operator* - A fuzzy operator (*AND* or *OR*) is used to combine the set of antecedents of each rule;
3. *Implication* - This method builds the consequential part of each rule;
4. *Defuzzification* - The aggregator and defuzzifier blocks to set a FIS are represented by the weighted sum operator.

A FIS allows a fast modeling of input-output relationships by extracting a set of rules able to model the nature of the data. Several variations/adaptations to the standard FIS process exist for improving the performance of the model. In particular, the centers and standard deviations of Gaussian membership functions can be adaptively modified to better fit the training data. In addition, an artificial neural network (ANN) can be trained to estimate the error even based on a small number of inputs used for the FIS [62]. Finally, a further stage may introduce also automatically new rules and corrections (based on expert knowledge) to the rules previously included in the FIS process.

In particular, neuro-fuzzy modeling exploits neural network-based learning techniques to have the FIS. In this case, a trial-and-error process, that stops when the desired accuracy is reached, is used to set decision rule sets. The fuzzy rule set is obtained by covering the input-output space of the samples with overlapped patches, where each patch represents a fuzzy rule. Note that a total coverage of samples is generally impossible to achieve since it is almost impossible to have a patch for each sample. In other words, the goal is to implement a technique that is as simple and efficient as possible for estimating the optimal number of clusters and the initial values of their centers in multivariate data. Then, from this information, an iterative optimization algorithm attempts to minimize a cost function that maintains the quality of the original data. Once the optimal configuration of clusters is selected then each cluster center is made to correspond to a fuzzy rule.

To implement the above process we used the GENFIS toolbox of Matlab [63]. This toolbox allows the extraction of fuzzy membership functions (FMFs) based on input/output pairs. Moreover, the Fuzzy Logic toolbox of Matlab include the *Adaptive Neuro-Fuzzy Inference System* (ANFIS) function [64, 65] for tuning the neuro-fuzzy inference system on the basis of some collections of input-output data. Therefore, a GENFIS+ANFIS approach was used in this work.

4. The Adopted Methodology

The above-described approach has been used to analyze users' decision criteria when choosing a transportation mode, which can be expressed by simple *if-then* rules fitting with fuzzy logic like:

if the time on mode M_1 **is** less than the time on mode M_2 **then** the mode M_1 is chosen

However, the decision processes underlying modal choices are usually more complex and more similar to:

if $t_{M_1} < t_{M_2}$ **and** $t_{ac/eg,M_1} < t_{ac/eg,M_2}$ **and** $K_{M_1} > K_{M_2}$ **and** \dots **then** the mode M_1 is chosen

which, in words, can be expressed in the form "if the travel time of M_1 is less than that of M_2 and the access/exit time of M_1 is less than that of M_2 and the comfort of M_1 is greater than that of M_2 and \dots then the user chooses the mode M_1 ".

In particular, in the random utility theory, a set of decision rules can be expressed as:

$$U_{M_1} = \beta_1 \cdot t_{M_1} + \beta_2 \cdot t_{ac/eg,M_1} + \beta_3 \cdot K_{M_1} + \dots + \epsilon_{M_1} \quad (1)$$

$$U_{M_2} = \beta_1 \cdot t_{M_2} + \beta_2 \cdot t_{ac/eg,M_2} + \beta_3 \cdot K_{M_2} + \dots + \epsilon_{M_2} \quad (2)$$

where the variables associated with the various transportation modes (e.g., t , $t_{ac/eg}$, K , etc.) are weighted by the set of parameters β obtained by a calibration procedure [9, 66]. In addition, a random term (ϵ) is also considered to take into account information and simplification in both model assumptions and mathematical formulations.

A further problem is the estimation of attributes referring to non-chosen transportation modes or qualitative attributes (e.g. comfort, etc.); in all these cases such attributes should be estimated by the analysts. Moreover, the number of variables to be taken into account in order to represent the behavior of users might be very large and then the calibration process could be time-consuming.

In the user's decision-making process some variables characterizing each transportation mode are relevant while other are just marginal. It is worthwhile to note that modeling user's behaviors should focus on identifying the set of variables that influence significantly the user's behavior. This aspect also meets one of the constraints of the FIS technique represented by the need to limit the number of variables in order to avoid a combinatorial explosion of the rule catalog. Therefore, the selection of the most representative variables (i.e., inputs) is an important stage with respect to both utility and FIS models.

In this work, a fuzzy curve approach was used to select the variables to be used as inputs to the FIS procedure, taking into account nonlinear effects [67, 68]. More in detail, with respect to a multiple-input single-output system (MISO), a travel mode is characterized by a set of variables x_i , with ($i = 1, \dots, m$) where m is the number of available travel modes, while y will denote the travel mode chosen by the user. For the k -th user and the t training patterns, the fuzzy curve c will be computed as:

$$c_i(x_i) = \frac{\sum_{k=1}^t y_k \cdot F_{i,k}(x_i)}{\sum_{k=1}^t F_{i,k}(x_i)} \quad \text{with } F_{i,k}(x_i) = e^{-\left(\frac{x_{i,k}-x_i}{s}\right)^2}$$

where $F_{i,k}$ is a Gaussian function $F_{i,k}(x_i)$, with x_i and s , respectively, being the mean and the standard deviation of the i -th coordinate of the k -th training pattern. This method consists in assessing the flatness of the fuzzy curve (c_i) because if it is too flat then the output will be weakly affected by inputs [51, 69].

The relevance of the input is determined based on a figure of merit, defined as the range of the fuzzy curve that is usually a fraction of the domain of the corresponding output variable (i.e., y) over the entire dataset of examples. More in detail, we subdivide the intervals of input and output in overlapped regions (consequently, also the fuzzy curve will be subdivided in overlapped regions) and label the corresponding value of the variable with fuzzy values. From each of the regions in which the fuzzy curve is partitioned it will be possible to derive a rule (with simple antecedent) that describes the input-output relationship for that region and, therefore, it will be possible to obtain a set of rules that approximates the fuzzy curve. In this way, the application of fuzzy patches (and thus fuzzy rules) is easier, although one must neglect possible correlations between input variables since there are as many fuzzy curves as the input-output pairs. Since there are no fuzzy curves for multiple simultaneous inputs, it is then necessary to use fuzzy surfaces to solve the problem. Fuzzy surfaces [70, 71] can be assumed as an extension of fuzzy curves. More formally, a fuzzy surface can be represented as:

$$c_i(x_i, x_j) = \frac{\sum_{k=1}^t y_k \cdot F_{i,k}(x_i) \cdot F_{j,k}(x_j)}{\sum_{k=1}^t F_{i,k}(x_i) \cdot F_{j,k}(x_j)} \quad \text{with } F_{i,k}(x_i) = e^{-\left(\frac{x_{i,k}-x_i}{s}\right)^2}$$

$$\text{and } F_{j,k}(x_j) = e^{-\left(\frac{x_{j,k}-x_j}{s}\right)^2}$$

In this case, the fuzzy rules have a double antecedent whose connective is *AND* since there are two inputs involved in the rule that approximates the fuzzy surface. Trivially, a fuzzy surface can generate countless fuzzy curves and for this reason extracting a catalog of rules directly from fuzzy surfaces is convenient. In fact, the use of fuzzy surfaces allows reducing the cardinality of the system (thanks to the presence of rules with double antecedent) and, moreover, the rules extracted from fuzzy surfaces trivially contain those extracted from fuzzy curves. Note that the use of fuzzy surfaces in the transportation domain is not new, for instance, in [72, 73] where a FIS leveraged to simulate and predict the future behavior of a vehicle by considering human factors of driving or in [74] to predict the level of congestion in heterogeneous networks by considering the flow and capacity of each arc of the network as input variables and the level of congestion as output. 75 rules and fuzzy surfaces have been employed to this purpose.

5. Modal Choice Module Validation

In this section we will show the performance of the modal choice module we designed. To this aim, a database obtained through the use of revealed preference techniques [9], has been used. The sample considered in the validation process consisted of 200 users for the training dataset and 500 users to realize the test dataset for home-work trips. The travel modes considered are walking (A), motorcycle (B), car-driver (C), car-passenger (D), and bus (E). Table 1 shows the percentages of users for the two different travel purposes and for the five travel modes.

The main variables considered in the surveys can be classified into; *i*) socio-economic variables (e.g., income, sex, age, private mode availability and so on) and *ii*) level of service variables (e.g., travel time, monetary costs, parking availability and so on) referred to the characteristics of the transportation network (see [9]). The input data are considered as fuzzy variables and each fuzzy variable, referred to a linguistic description, is characterized by a measure of membership in each of the considered linguistic properties.

In particular, the steps of the process to model a FIS are:

1. *Inputs Fuzzification* - Here the inputs membership degree to the relevant fuzzy sets are determined. The input is always a crisp value ranging in the domain of the input variable, while the output is a fuzzy degree of membership in the $[0;1]$ domain. Consequently, we know the belonging degree of each antecedent with respect to each rule.
2. *Fuzzy Operators Application* - The fuzzy operator is applied to obtain a representative value of the antecedent result (simple or composed) for that rule to be applied to the output functions. The input fuzzy operator consists of two or more membership values derived from the fuzzification of the input variables, while the associated output consists of a single truth value.
3. *Outputs Aggregation* - In this step, the fuzzy outputs produced by each rule are unified into a single fuzzy set, ready to be defuzzified, by graphical superposition. In the aggregation process, the input is the set of membership functions (stopped or scaled), resulting from the implication process on each rule, while the output is represented by a fuzzy set for each output variable.
4. *Output Defuzzification* - In input this process receives the aggregated fuzzy outputs and in output returns a crisp value obtained by calculating the barycenter of the geometrical

Travel Modes	Code	%
Walking	A	10.8
Motorcycle	B	9.2
Car-driver	C	56.4
Car-passenger	D	6.2
Bus	E	17.4

Table 1

Percentage of transport mode usage for the exploited datasets.

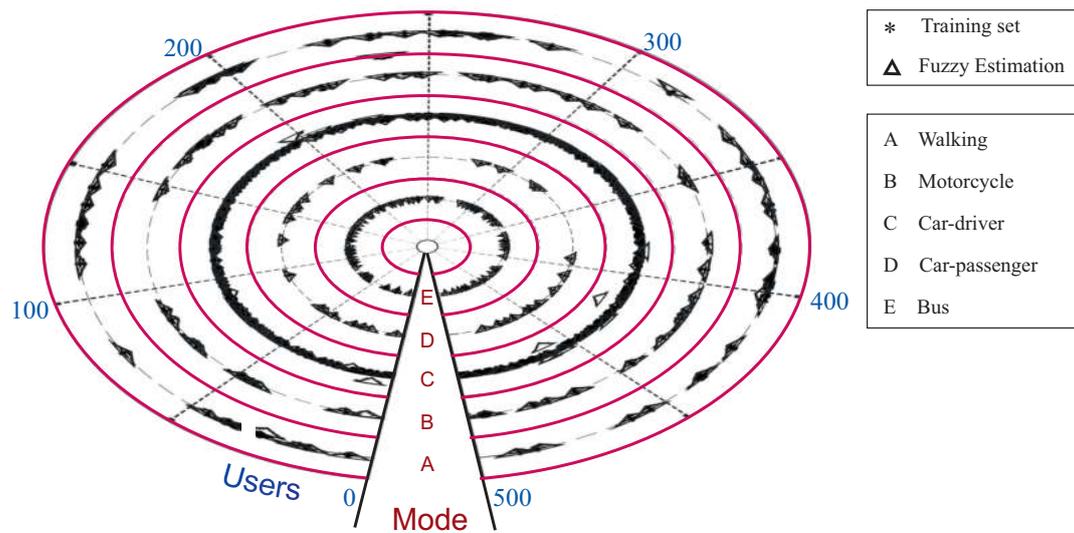


Figure 2: Accuracy of the neuro-fuzzy approach in predicting users' mode choices.

shape derived from the aggregation of the outputs. Moreover, by adopting Sugeno's FIS, peaks (or singletons) are used as consequents then simplifying the defuzzification process. Accordingly, the main features of the FIS generated by GENFIS can be summarized in 12 inputs representing the qualitative variables reported in the database used for the validation of the module (i.e., the attributes of the travel modes available to users such as time, monetary costs, vehicle ownership, saturation ratio of parking facilities at destination, etc.), 16 rules and a single output for each user, represented by the chosen travel mode.

Based on the input-output patterns, a conventional FIS was first designed without making use of any learning techniques and, as expected, unsatisfactory performance have been obtained. Subsequently, the toolbox GENFIS and the function ANFIS were used to improve the performance of the initial FIS. In particular, the neural network used by ANFIS is made by five layers, namely, the first layer for the input, the second layer for the membership functions associated with each input, the third layer for the implication constructs, the fourth layer for the aggregation operation and, finally, the fifth layer for the output. Note that the membership functions, after learning, may be overlapped even in significant numbers

The predictions calculated after the training phase by applying the proposed methodology to the test dataset for home-work trips are represented in Figure 2, for both the users and the various transport modes. Each marker identifies the prediction accuracy between a user's choice and a transport mode (identified from A to E), while not aligned markers identify incorrect predictions. In other words, the closer the markers are to the red line, the worse is the prediction; conversely, the closer the markers are to the dotted line (where each dotted line identifies a transportation mode), the more accurate the prediction is. From Figure 2, it is evident that the proposed neuro-fuzzy approach has a high degree of accuracy in correctly predicting the

transport mode chosen by the user in most cases. The neuro-fuzzy approach is wrong only in presence of users characterized by very specific attributes (which can be considered as outliers). The high level of accuracy indicates the capability of the rules calibrated by the model to simulate the real behavior of users, thus confirming the basis of many economic theories that the behavior of users can be inferred from their choices. Further analyses of the results show that users primarily give a high value to travel time with respect to, for example, travel costs and prefer private transport modes such as cars and motorcycles, except when the ratio of vehicles in the family versus the number of its members is much less than 1.

Although the degree of accuracy of the modal choice predictions is satisfactory, further refinements (mainly in terms of better understanding users' behavior) would be possible if additional qualitative information such as user information systems, comfort, privacy, etc., not present in the dataset used, were available.

Note that with quantitative data, the two methods, random utility models and fuzzy methods, arrive at the same results in terms of predictions, but differ on the basis of the specific information provided (e.g., semantic rules versus parameter values). Differently, the presence of qualitative variables does not change the operation and predictions produced by the proposed neuro-fuzzy approach (e.g., semantic rules are considered with both quantitative and qualitative variables), while random utility models require a conversion from qualitative to quantitative variables to be operated on the basis of a conversion scale. In the latter case, it is evident how the correctness of the predictions is affected by the conversion scale adopted. Finally, the feedback released in the last step (see Section 2) should ensure the agent's ability to adapt its mode choices over time as conditions change.

6. Conclusions

This paper proposed a neuro-fuzzy model to feed a more complete agent-based structure where travel choice behavioral rules are embedded. More in detail, the paper has focused on the learning step of user agents, based on fuzzy logic approaches, to recognize the most important rules taken into account by users when they choose a transport mode.

Among the relevant features of fuzzy logic approaches, there are the use of linguistic variables either in place or in addition to numerical variables; the identification of associations among variables by using fuzzy conditional statements; the implementation of complex relationships by using fuzzy-based algorithms.

In this work, experimental data have been used to find linguistic rules in the form *if-then* whose antecedents and consequents utilize fuzzy sets instead of crisp numbers. The fuzzy inference models derived from the previous rules allow understanding user's mode choice behavior and provide user agents with such knowledge. Particularly, behavioral rules make user agents not only able to simulate traveler's choices but also to modify their choices according to such rules in case of variations in the transportation system – mainly level of service variable variations.

The main advantages of the fuzzy approach are: *i*) opportunity to obtain rules with direct interpretation; *ii*) simple use of such results also by generalist analysts; *iii*) opportunity to improve the model by adding further information of the experts in the field (expert judgments).

In addition, the fuzzy model does not require high computing complexity, particularly for the on-line applications, and its “network” structure facilitates the implementation of a hardware system with relatively little costs. The main disadvantage is the limited number of inputs required by the fuzzy model to work efficiently, which implies data-compression techniques and suitable reduction of the inputs.

The results obtained in this study are very promising, not only they are comparable with other known approaches (e.g., random utility models), but they offer greater opportunity for simulating user behavior in a more complete user agent framework addressed to simulate transportation systems. Particularly, feedback obtained at the last step - as a result of the interactions with the *Transportation Networks Module* in the user agent framework - might be used to feed learning steps, which could be improved continuously.

Finally, as a further development, we note how the performance of the designed *Mode Choice Module* makes it also suitable to be embedded in a personal agent assistant to support users in their daily travel activities

References

- [1] C. Ribeiro, J. V. Ferreira, P. Partidário, Life cycle assessment of a multi-material car component, *The International Journal of Life Cycle Assessment* 12 (2007) 336–345.
- [2] C. Willing, T. Brandt, D. Neumann, Intermodal mobility, *Business & Information Systems Engineering* 59 (2017) 173–179.
- [3] E. Pisoni, P. Christidis, P. Thunis, M. Trombetti, Evaluating the impact of “sustainable urban mobility plans” on urban background air quality, *Journal of environmental management* 231 (2019) 249–255.
- [4] J. van Zeben, Environmental implications of the eu’s urban mobility agenda, in: *Smart Urban Mobility*, Springer, 2020, pp. 41–59.
- [5] R. Faria, L. Brito, K. Baras, J. Silva, Smart mobility: A survey, in: *2017 International Conference on Internet of Things for the Global Community (IoTGC)*, IEEE, 2017, pp. 1–8.
- [6] F. Pinna, F. Masala, C. Garau, Urban policies and mobility trends in italian smart cities, *Sustainability* 9 (2017) 494.
- [7] J. Zawieska, J. Pieriegud, Smart city as a tool for sustainable mobility and transport decarbonisation, *Transport Policy* 63 (2018) 39–50.
- [8] M. Miskolczi, D. Földes, A. Munkácsy, M. Jászberényi, Urban mobility scenarios until the 2030s, *Sustainable Cities and Society* (2021) 103029.
- [9] M. N. Postorino, A comparative analysis of different specifications of modal choice models in an urban area, *European journal of operational research* 71 (1993) 288–302.
- [10] Y. Tyrinopoulos, C. Antoniou, Factors affecting modal choice in urban mobility, *European Transport Research Review* 5 (2013) 27–39.
- [11] V. Pattanaik, M. Singh, P. Gupta, S. Singh, Smart real-time traffic congestion estimation and clustering technique for urban vehicular roads, in: *2016 IEEE region 10 conference (TENCON)*, IEEE, 2016, pp. 3420–3423.
- [12] E. Anagnostopoulou, J. Urbančič, E. Bothos, B. Magoutas, L. Bradesko, J. Schrammel, G. Mentzas, From mobility patterns to behavioural change: leveraging travel behaviour

- and personality profiles to nudge for sustainable transportation, *Journal of Intelligent Information Systems* 54 (2020) 157–178.
- [13] E. Cascetta, M. N. Postorino, Fixed point approaches to the estimation of o/d matrices using traffic counts on congested networks, *Transportation science* 35 (2001) 134–147.
- [14] M. N. Postorino, G. Musolino, P. Velonà, Evaluation of o/d trip matrices by traffic counts in transit systems, in: *Schedule-Based Dynamic Transit Modeling: theory and applications*, Springer, 2004, pp. 197–216.
- [15] X. Ros Roca, L. Montero Mercadé, J. Barceló Bugada, Transport analytic approaches to the dynamic origin-destination estimation problem, in: *Congreso CAMPUS FIT 2020: online: 24-26 junio, 2020: libro de actas*, Universitat Politècnica de Catalunya., 2020, pp. 41–57.
- [16] M. Fosgerau, E. Frejinger, A. Karlstrom, A link based network route choice model with unrestricted choice set, *Transportation Research Part B: Methodological* 56 (2013) 70–80.
- [17] M. Patriksson, *The traffic assignment problem: models and methods*, Courier Dover Publications, 2015.
- [18] M. N. Postorino, V. Fedele, The analytic hierarchy process to evaluate the quality of service in transit systems, *WIT Transactions on The Built Environment* 89 (2006).
- [19] B. Lenz, E. Fraedrich, New mobility concepts and autonomous driving: the potential for change, in: *Autonomous driving*, Springer, 2016, pp. 173–191.
- [20] E. Picasso, M. N. Postorino, G. M. L. Sarné, A study to promote car-sharing by adopting a reputation system in a multi-agent context., in: *WOA, 2017*, pp. 13–18.
- [21] G. M. Becker, M. H. DeGroot, J. Marschak, Stochastic models of choice behavior, *Behavioral science* 8 (1963) 41–55.
- [22] C. F. Manski, The structure of random utility models, *Theory and decision* 8 (1977) 229.
- [23] J. Walker, M. Ben-Akiva, Generalized random utility model, *Mathematical social sciences* 43 (2002) 303–343.
- [24] J. J. Louviere, D. A. Hensher, J. D. Swait, *Stated choice methods: analysis and applications*, Cambridge university press, 2000.
- [25] A. Dorri, S. S. Kanhere, R. Jurdak, Multi-agent systems: A survey, *Ieee Access* 6 (2018) 28573–28593.
- [26] M. P. Keane, A note on identification in the multinomial probit model, *Journal of Business & Economic Statistics* 10 (1992) 193–200.
- [27] D. A. Hensher, W. H. Greene, Specification and estimation of the nested logit model: alternative normalisations, *Transportation Research Part B: Methodological* 36 (2002) 1–17.
- [28] M. J. Gaudry, M. G. Dagenais, The dogit model, *Transportation Research Part B: Methodological* 13 (1979) 105–111.
- [29] G. E. Box, D. R. Cox, An analysis of transformations, *Journal of the Royal Statistical Society: Series B (Methodological)* 26 (1964) 211–243.
- [30] M. J. Gaudry, The inverse power transformation logit and dogit mode choice models, *Transportation Research part B: Methodological* 15 (1981) 97–103.
- [31] M. Bierlaire, A theoretical analysis of the cross-nested logit model, *Annals of operations research* 144 (2006) 287–300.
- [32] C.-H. Wen, F. S. Koppelman, The generalized nested logit model, *Transportation Research Part B: Methodological* 35 (2001) 627–641.

- [33] E. Cascetta, A. Papola, Dominance among alternatives in random utility models, *Transportation Research Part A: Policy and Practice* 43 (2009) 170–179.
- [34] J.-S. Jang, C.-T. Sun, Neuro-fuzzy modeling and control, *Proceedings of the IEEE* 83 (1995) 378–406.
- [35] R. Babuška, H. Verbruggen, Neuro-fuzzy methods for nonlinear system identification, *Annual reviews in control* 27 (2003) 73–85.
- [36] P. V. de Campos Souza, Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature, *Applied soft computing* 92 (2020) 106275.
- [37] M. N. Postorino, G. M. L. Sarné, Agents meet traffic simulation, control and management: A review of selected recent contributions, in: *Proceedings of the 17th Workshop “from Objects to Agents”, WOA*, volume 1664, 2016, pp. 112–117.
- [38] K. Bernhardt, Agent-based modeling in transportation, *Artificial Intelligence in Transportation* 72 (2007).
- [39] G. O. Kagho, M. Balac, K. W. Axhausen, Agent-based models in transport planning: Current state, issues, and expectations, *Procedia Computer Science* 170 (2020) 726–732.
- [40] D. Pamucar, G. Ćirović, Vehicle route selection with an adaptive neuro fuzzy inference system in uncertainty conditions, *Decision Making: Applications in Management and Engineering* 1 (2018) 13–37.
- [41] N. Khademi, M. Rajabi, A. S. Mohaymany, M. Samadzad, Day-to-day travel time perception modeling using an adaptive-network-based fuzzy inference system (anfis), *EURO Journal on Transportation and Logistics* 5 (2016) 25–52.
- [42] K. Kravari, N. Bassiliades, A survey of agent platforms, *Journal of Artificial Societies and Social Simulation* 18 (2015) 11.
- [43] S. Abar, G. K. Theodoropoulos, P. Lemarinier, G. M. O’Hare, Agent based modelling and simulation tools: A review of the state-of-art software, *Computer Science Review* 24 (2017) 13–33.
- [44] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, Y. Liu, A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges, *IEEE Communications Surveys & Tutorials* 21 (2018) 393–430.
- [45] E. Bonabeau, Agent-based modeling: Methods and techniques for simulating human systems, *Proceedings of the national academy of sciences* 99 (2002) 7280–7287.
- [46] R. J. Allan, et al., *Survey of agent based modelling and simulation tools*, Science & Technology Facilities Council New York, 2010.
- [47] O. Kamara-Esteban, G. Azkune, A. Pijoan, C. E. Borges, A. Alonso-Vicario, D. López-de Ipiña, Massha: an agent-based approach for human activity simulation in intelligent environments, *Pervasive and Mobile Computing* 40 (2017) 279–300.
- [48] S. Bandini, S. Manzoni, G. Vizzari, Agent based modeling and simulation: an informatics perspective, *Journal of Artificial Societies and Social Simulation* 12 (2009) 4.
- [49] K. Andrade, K. Uchida, S. Kagaya, Development of transport mode choice model by using adaptive neuro-fuzzy inference system, *Transportation Research Record* 1977 (2006) 8–16.
- [50] M. N. Postorino, M. Versaci, A neuro-fuzzy approach to simulate the user mode choice behaviour in a travel decision framework, *International Journal of Modelling and Simulation* 28 (2008) 64–71.

- [51] M. N. Postorino, M. Versaci, Modelling user mode choices by an ellipsoidal fuzzy approach, *International Journal of Modelling and Simulation* 33 (2013) 235–243.
- [52] S. Minal, C. R. Sekhar, E. Madhu, Development of neuro-fuzzy-based multimodal mode choice model for commuter in delhi, *IET Intelligent Transport Systems* 13 (2019) 243–251.
- [53] M. N. Postorino, G. M. L. Sarné, Reinventing mobility paradigms: Flying car scenarios and challenges for urban mobility, *Sustainability* 12 (2020) 3581.
- [54] M. N. Postorino, F. A. Sarné, G. M. L. Sarné, An agent-based simulator for urban air mobility scenarios., in: *WOA*, volume 1613, 2020, p. 2.
- [55] J. D. Pineda-Jaramillo, A review of machine learning (ml) algorithms used for modeling travel mode choice, *Dyna* 86 (2019) 32–41.
- [56] L. A. Zadeh, Fuzzy sets, in: *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*, World Scientific, 1996, pp. 394–432.
- [57] S. Guillaume, Designing fuzzy inference systems from data: An interpretability-oriented review, *IEEE Transactions on fuzzy systems* 9 (2001) 426–443.
- [58] B. Kosko, *Fuzzy engineering*, Prentice-Hall, Inc., 1996.
- [59] S. Roychowdhury, W. Pedrycz, A survey of defuzzification strategies, *International Journal of intelligent systems* 16 (2001) 679–695.
- [60] M. Sugeno, An introductory survey of fuzzy control, *Information sciences* 36 (1985) 59–83.
- [61] Z.-H. Xiu, G. Ren, Stability analysis and systematic design of takagi–sugeno fuzzy control systems, *Fuzzy sets and systems* 151 (2005) 119–138.
- [62] J. Vieira, F. M. Dias, A. Mota, Neuro-fuzzy systems: a survey, in: *5th WSEAS NNA international conference on neural networks and applications*, Udine, Italia, 2004, pp. 87–92.
- [63] Matlab, <https://matlab.mathworks.com>, 2021.
- [64] J. Jang, R. ANFIS, Adaptive network-based fuzzy inference system, *IEEE Trans on Systems, Man and Cybernetics* 23 (1993) 665–685.
- [65] N. Walia, H. Singh, A. Sharma, Anfis: Adaptive neuro-fuzzy inference system-a survey, *International Journal of Computer Applications* 123 (2015).
- [66] Y. E. Hawas, Development and calibration of route choice utility models: neuro-fuzzy approach, *Journal of transportation engineering* 130 (2004) 171–182.
- [67] C. R. Bhat, Covariance heterogeneity in nested logit models: econometric structure and application to intercity travel, *Transportation Research Part B: Methodological* 31 (1997) 11–21.
- [68] C. R. Bhat, F. S. Koppelman, Activity-based modeling of travel demand, in: *Handbook of transportation Science*, Springer, 1999, pp. 35–61.
- [69] A. W. Sadek, G. Spring, B. L. Smith, Toward more effective transportation applications of computational intelligence paradigms, *Transportation research record* 1836 (2003) 57–63.
- [70] Y. Lin, G. A. Cunningham, S. V. Coggeshall, R. D. Jones, Nonlinear system input structure identification: Two stage fuzzy curves and surfaces, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 28 (1998) 678–684.
- [71] K. Shihabudheen, G. N. Pillai, Recent advances in neuro-fuzzy system: A survey, *Knowledge-Based Systems* 152 (2018) 136–162.
- [72] A. Khodayari, R. Kazemi, A. Ghaffari, R. Braunstingl, Design of an improved fuzzy logic based model for prediction of car following behavior, in: *2011 IEEE International*

- Conference on Mechatronics, IEEE, 2011, pp. 200–205.
- [73] A. Ghaffari, A. Khodayari, A. Kamali, F. Tajdari, N. Hosseinkhani, New fuzzy solution for determining anticipation and evaluation behavior during car-following maneuvers, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering* 232 (2018) 936–945.
- [74] M. Amini, M. F. Hatwagner, G. C. Mikulai, L. T. Koczy, An intelligent traffic congestion detection approach based on fuzzy inference system, in: *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, IEEE, 2021, pp. 97–104.

User Assistance for Predicting the Availability of Bikes at Bike Stations

Claudia Cavallaro¹, Emiliano Tramontana²

¹INFN-CNAF Centro Nazionale per la Ricerca e lo Sviluppo nelle Tecnologie Informatiche e Telematiche, Viale Berti Pichat 6/2, 40127 Bologna, Italy.

²Department of Mathematics and Computer Science, University of Catania, Viale Andrea Doria, 6, 95125 Catania, Italy.

Abstract

Generally, assistant agents have been employed to recommend users some goods or services they could be interested in. Moreover, thanks to devices recording user geographical positions, recommendation systems have been developed to propose places to visit. This paper proposes an approach for making an estimate of bikes available at bike stations, hence facilitating the use of such a transport means. I.e. users made aware of bike availability beforehand can choose such a transport mode more easily. By using a fast algorithm analysing data recording all bike movements, we obtain an accurate estimate of where bikes will be in the near future. This is possible by determining the frequent paths of bikes, according to their starting points, and the likely destinations. Thanks to such estimates, users can be alerted beforehand about the desired bike at their preferred bike station. Then, a message will be sent when the bike is in the station. Assistant agents giving such alerts are provided to users as an app on their smartphone.

Keywords

Assistant agents, Recommendation system, Bike-sharing, Forecast, People gathering

1. Introduction

Plenty of data are available from, and are regularly produced by, personal devices tracking the movement of people in terms of GPS coordinates. Such data gathered from devices, as e.g. smartphones, smart watches, cars, are usually sent to an aggregator host and then used to gain knowledge on the the behaviour of people, on possible routes, on the density of people in some places, etc. [1, 2, 3]. E.g. it is possible to determine the most visited tourist places in a city and the usual paths traversed to reach such places [4]. Apps using data gathered from the movement of people have been offered to users to suggest lively places to visit, e.g. in a certain time-frame [5, 6], paths inducing some kinds of emotions [7, 8], etc. An app on a smartphone, while contributing to data gathering, provides users with a means to be notified of potential useful information, such as e.g. a nearby point of interest. By giving users suggestions, and acting on behalf of the user for finding the best spot, by communicating with a server, such an app can be seen as an *agent* assisting the user in her daily routine.

WOA 2021: Workshop “From Objects to Agents”, September 1–3, 2021, Bologna, Italy

✉ claudia.cavallaro@cnafe.infn.it (C. Cavallaro); tramontana@dmi.unict.it (E. Tramontana)

🆔 0000-0003-3938-0947 (C. Cavallaro); 0000-0002-7169-659X (E. Tramontana)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

In previous studies, the analysis of GPS coordinates has been performed by using several techniques, such as firstly computing the point of interests, and then determining how frequently each user can be found nearby. Apriori algorithm has been used to find how frequently a set of places have been visited by users passing through the same path [6]. However, the execution of the Apriori algorithm takes a long time [9]. Apriori has a complexity of $O(2^n)$, where n is the number of places to be considered. Hence, such a complexity requires a great deal of computation.

Sure, for analysing the ever growing amount of GPS coordinates recorded, it is paramount to have a solution that minimizes computation time. In this work, we consider the position of bikes in a bike sharing scenario and compute the common paths and the probable destinations with a certain probability. This is useful to give users that wish to rent a bike the availability of bikes at bike stations in the near future. Moreover, should restrictions on the gathering of people in public spaces persist (due to the pandemic outbreak), thanks to the available estimate on the number of bikes arriving at a bike station, the devised app can be handy to give people an alert beforehand suggesting the best time for approaching the bike station.

This paper proposes a novel approach to process data concerning the geographic positions of bikes, to find common paths and infer possible destinations. The used algorithm is FP-Growth that has been shown as having a lower execution time, when compared with Apriori [10, 9], as FP-Growth is a linear time algorithm¹. This approach is novel as FP-Growth has never been applied for the said objective before.

The rest of the paper has the following structure. Section 2 gives the comparison with the relevant related work. Section 3 describes the proposed software architecture to assist users in finding bike stations with available bikes. Section 4 introduces the dataset used for experiments. Section 5 details how the FP-Growth algorithm has been used for geo-spatial data. Section 6 shows the results when analysing a big amount of data related to bike positions with the proposed approach. Finally, Section 7 reports our concluding remarks.

2. Related works

The Spatio-Temporal Data Mining problem has been addressed in many works in the literature by using different techniques. This section presents some of the most relevant approaches having similarities with respect to our approach.

In [11], Pensa et al. addressed the problem of finding frequent sequential patterns for a dataset of real vehicle GPS trajectories tracked in Milan, Italy. The proposed algorithm transforms spatio-temporal trajectories into sequences of regions of interest (ROI), based on a discretization of the working space through a regular grid. The authors measured the similarity of each pair of patterns as support density and length. By using the technique PrefixSpan [12], ROI's sequences were represented as a tree, where each node is a ROI. Then, each branch of the tree was associated with the support in each trajectory and subtrees having infrequent ROIs were removed. Finally, they presented the framework P2kA that anonymizes the dataset and thus preserves the privacy of users in the frequent paths extracted. Unlike our proposal, in [11] the objective was not aiming at making predictions on future destinations, while our strategy

¹<https://www.softwaretestinghelp.com/fp-growth-algorithm-data-mining/>

allows us, with a certain probability, to identify the next bike station reached by the same group of cyclists racing together. Moreover, our work need not obscure certain information on the data, because the ID associated with a registered route refers to a specific rented bike in a time slot and not to the customer's card.

Other data mining methods, such as DBSCAN [13], were used for the purpose of identifying parts of common routes or shared Points Of Interest [4]. In [14], Crociani et al. offered an unsupervised learning approach for an automatic lane detection in multidirectional pedestrian flows, taking into account the angular distance during the movement.

Regarding recommendation systems, in [7], of Quercia et al. gathered metadata from the pictures in Flickr, and determined routes to be suggested according to parameters of beauty, quietness and happiness. They computed the probability that an individual visits a certain destination because it is pleasant. In the literature other algorithms, such as the one presented in [15], recommended public transport routes that include both short walks and reduced waiting times. Our forecast indicates how many cyclists of a given group will subsequently head to a certain point, according to previous traffic data. This aims to: (i) show bikes availability, and (ii) alert users in case some places will be overcrowded. Another travel recommendation system that analyzes GPS trajectories was presented in [16]: the authors used HITS [17] algorithm, and recommendations were based on the travel experiences of the users (hub scores) and the interests of the road segments (authority scores).

In [18], Cecaj and Mamei presented a useful way to extract associative rules, with the Eclat algorithm, to find co-locations of companies in industrial agglomerations. Our approach mines the frequent stations from bike trips and differs for the purpose of forecasting future displacements. Some differences between three frequent mining pattern algorithms (Apriori, Eclat and FP-Growth) are shown extensively in [19].

3. Multi-Agent Architecture

Our proposed system comprises an app running on a smartphone and acting as an assistant *agent* for the user. Such an app communicates with a server side both to send gathered data, i.e. geographical coordinates, as well as user requests. Moreover, the app receives useful data that the server has gained from data analysis and selected as possibly useful to the user according to her preferences and requests.

In our application, the user is interested in finding a given type of bike available in one among a few nearby bike stations. Hence, according to our approach, the server performs data analysis to estimate the future availability of bikes and sends to interested users, on their app, an alert when the bike is likely to be available and when. The interested user, by means of the app can express the preference to lock the bike, then and the server will give her confirmation of availability. Figure 1 shows a sketch of some dialog panels a user can be provided with: from the left there is an opening logo, then a screen for letting the user input her preferences on the bike she wishes to rent, then a screen for the preferred bike stations, and finally a screen showing the number available bikes at the selected bike stations. The user preferences are sent from the mobile app (assistant agent) to a server, whereas the alert and details on bikes available are received from the server on the mobile app.



Figure 1: User interface provided by the app on the smartphone by the assistant agent.

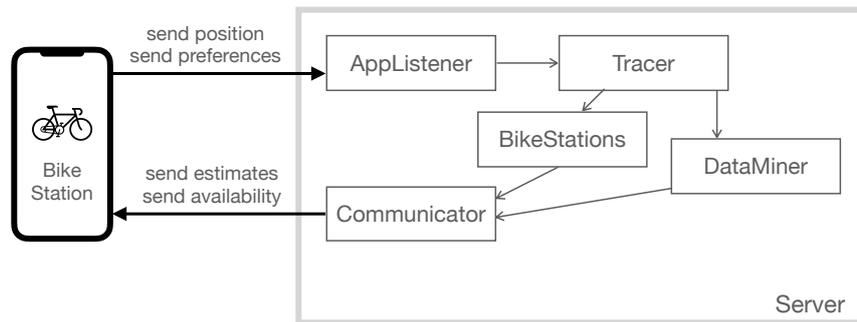


Figure 2: Software architecture for the multi-agent system.

Agents, as apps on a smartphone, communicate indirectly to each other, i.e. they rely on the server to send data, as their location or trajectory when having rented a bike, since such data are useful to make forecast of future destinations, hence give estimate of bike availability to others. The indirect communication between agents ensures the privacy of users, as a user location and identity cannot be discovered by other users. The server collects the ID of the bikes and the position, not the identity of the user, hence the safety of users is enhanced. Moreover, a user can set to have her coordinates being altered by a random value within a range to make it uncertain the exact position even to the server further enhancing the privacy of the user [20].

Figure 2 gives an overview of the software architecture. The app on the smartphone communicates with a server side, which is a host where a component called *AppListener* handles incoming data transmitted by each app (therefore by each user). The data are: the bike positions sent periodically, and user preferences, i.e. the type of bike, the station of interest, rental time, etc. The *AppListener* component sends the position data to the *Tracer*, and the latter updates the *DataMiner* with new data periodically. The data on the stations and the bikes present in the stations are kept by *BikeStations*, and updated according to the positions of the bikes. *DataMiner* implements the proposed estimation algorithm (explained in the following sections). When there is some significant data to communicate to the user, this is transmitted from the server side to the app via the *Communicator* component.

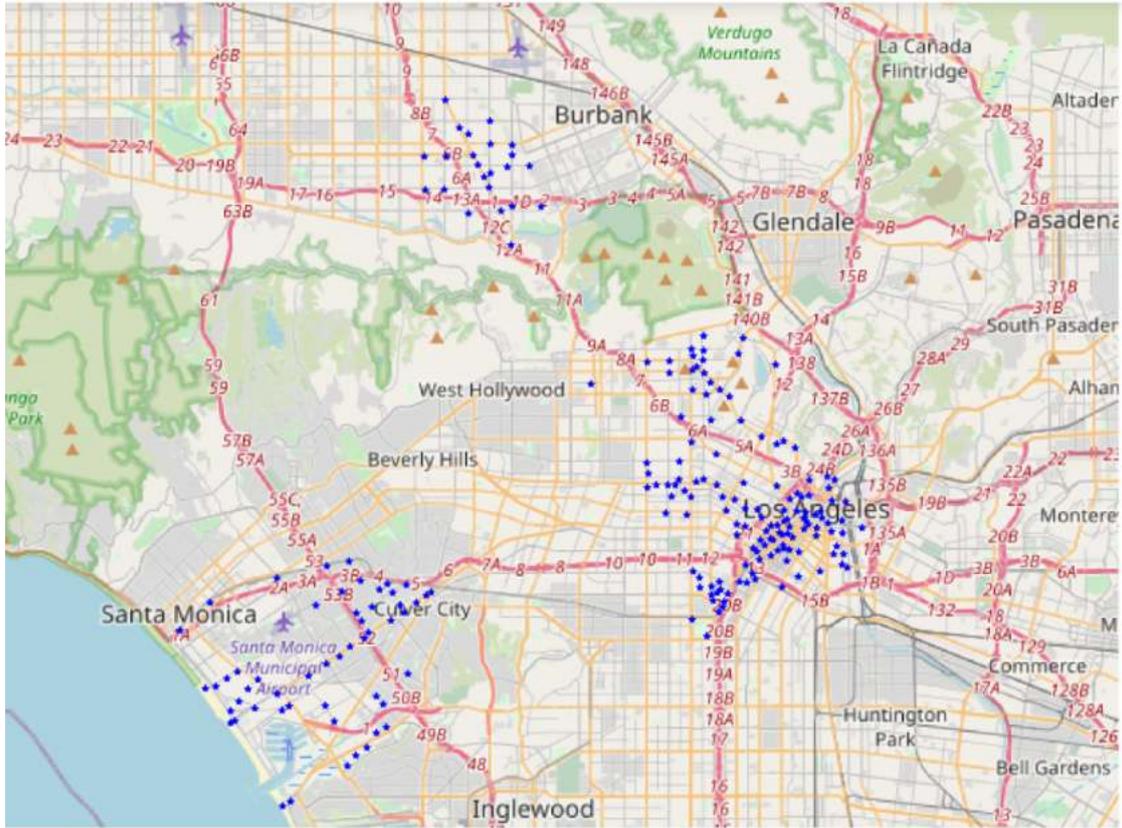


Figure 3: Bike stations are represented by blue stars and relate to trips recorded in the period April-June 2021. [Google Maps, accessed July, 2021]

4. Dataset

The dataset used in our experiments to perform tests was provided by Metro Bike Share². Cyclist trips in Los Angeles, California, have been collected since July 7, 2016, and this project includes recordings for the following 20 quarters. The file of each period, for each line, includes: a numeric travel identifier, travel duration in minutes, travel start date and time, start station name with its geographical position (latitude, longitude), end station name and its geographical position (latitude, longitude), and the numeric identifier of the bike. Moreover, there are pass subscription details, and bike type.

The position of each person who uses the bike sharing service is identified each time she uses the card in the bike parking stations. User privacy is guaranteed since a numeric identifier is associated with each bike used. Journeys lasting less than one minute and trial journeys are not retained. At present the dataset corresponds to 1,252,386 registered trips of 208,000 users, for a total of 4,091,563 miles.

²<https://bikeshare.metro.net/about/data/>

5. Methodology

In this section we present a description of FP-Growth, a data mining algorithm proposed by Han et al. [21, 10]. We highlight that it is our novel proposed idea to apply it for the geo-spatial field in order to find common paths using an algorithm having a short execution time. Then as a further step in our proposal we predict future movements.

The common purpose of this and other data mining algorithms, and among the most interesting we mention Apriori [22], is to extract frequent elements or tuples of elements from a set of transactions and the association rules from large datasets. It is possible to find more details of the application of Apriori algorithm in order to find common paths in [23], and as far as the prediction of movements in a discretized spatial area in [6]. A disadvantage of this strategy, however, is that the count of the *support* of an element (i.e. the number of transactions it is present in) requires scanning the dataset over and over again, and this greatly affects the execution time.

The main difference that led us to opt for the FP-Growth algorithm is that the generation of frequent candidates is not necessary for it. This was done in Apriori at each iteration, that is, every time an element of greater size than that of the previous level was investigated. FP-Growth works on a tree representation of the starting dataset, therefore, thanks to this more compact storage, managing large volumes of data is no longer a problem. Memory errors due to large occurrence tables, necessary for Apriori processing, are thus overcome.

Each set of elements is mapped into a specific path of the tree and the efficiency of this algorithm derives from its “divide et impera” method. The initial part of this approach is to find the support of each element, in our case, to determine which stations are shared by a chosen minimum number of users. After finding these frequent points, they are sorted in descending order according to their support and associated with a branch of the tree. Figures 4 and 5 show how FP-Growth works using a small-sized example.

The tree is then updated: (i) if an element already exists in it, its count is increased, otherwise

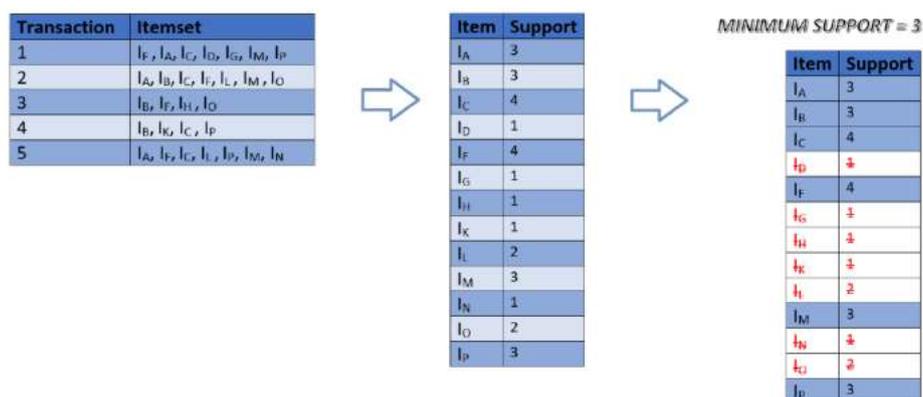


Figure 4: How FP-Growth works using a small example: on the left a list of transactions and their items, in the centre a list of items and its support, on the right the selection of items having a minimum given value of the support.

Frequent item	Support
I_F	4
I_C	4
I_A	3
I_B	3
I_M	3
I_P	3

Transaction	Itemset
1	$I_F, I_A, I_C, I_D, I_G, I_M, I_P$
2	$I_A, I_B, I_C, I_F, I, I_M, I_G$
3	I_B, I, I_M, I_G
4	I_B, I_C, I_G, I_P
5	$I_A, I_F, I_C, I, I_B, I_M, I_U$

Transaction	Ordered Itemset, with support ≥ 3
1	I_F, I_C, I_A, I_M, I_P
2	I_F, I_C, I_A, I_B, I_M
3	I_F, I_B
4	I_C, I_B, I_P
5	I_F, I_C, I_A, I_M, I_P

Figure 5: Given the example shown in Figure 4, the right table shows the transactions having the items whose support is at least a given value.

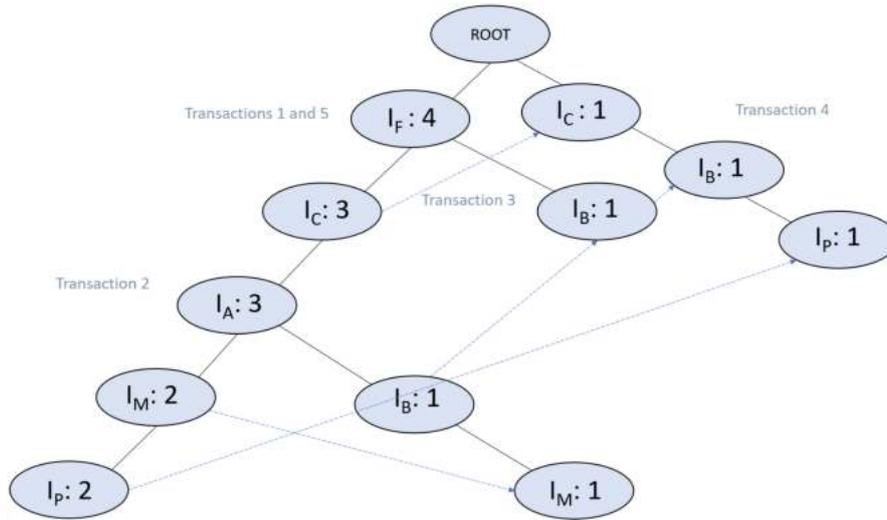


Figure 6: The FP-Tree of the example in Figures 4 and 5.

(ii) a new branch is created and connected to the parent node. To find a common path, starting from an element of the root, we scroll along a branch of the tree. Further examples and details of FP-Growth, even though not applied to the geo-spatial context, can be found in [24].

Figure 6 shows the final FP-Tree of the previous example. In this case the frequent itemsets extracted are: $I_F - I_C - I_A - I_M$ with support = 3, and all its subsets with support 3, except $I_F = 4$; $I_C - I_P$ with support = 3, I_C with support 4, I_P with support = 3 and I_B with support = 3.

Note that the FP-Growth runtime increases linearly with the number of elements, while in Apriori this growth is exponential. The frequent patterns in Apriori are obtained after the execution of all the iterations, while with FP-Growth it is possible to obtain them starting from the root-element and scrolling along a single branch. The dataset scan with FP-Growth method is done only twice, as opposed to Apriori in which the scan is done as many times as the number of iterations. Sorting by decreasing frequency allows a faster execution time than for the increasing one.

We can say that FP-Growth is the best strategy for our goal, which allows us to effectively

find common routes among the same group of cyclists in a large (spatial) dataset. According to our approach, in fact, it is not necessary to make a comparison of trips (sequences of stops in bike parking) for all pairs of users. Our experiments show that this method is about 6 times faster than Apriori.

6. Results

This section shows the results of our tests. For the bike dataset, in the most recent quarter (April-June 2021) the data consist of 59,081 routes, 2,824 cyclists traveling between 215 bike start stations and 215 end stations, and these are shown in the map drawn in Figure 3.

By applying the FP-Growth algorithm in order to find the routes connecting two “hot” stations, we initially set a high support: corresponding to 10% of cyclists in different time slots. The result showed us 8 pairs of common stations shared by a minimum of 282 cyclists to a maximum of 442. The execution time of the FP-Growth algorithm was 40 *ms* on an Intel Core i5 at 1GHz having 16 GB of RAM: all the tests were performed in Python 3 language.

By lowering the threshold of number of users, for a support equal to 100 bike IDs, 153 different routes were detected in a run time of 116 *ms*. To find the small groups of people moving together, we set the minimum support as 5 by running the algorithm on the columns of the dataset corresponding to: same departure station, same arrival station, same departure time and same end time. The test showed groups of 5 to 7 cyclists who shared 13 routes in the same time slot (9am-12pm). Since the exact position of cyclists in the intermediate roads between two stations is not known, to verify that the common routes are plausible we checked the value in the “duration trips” column, which was compatible for the whole group that was moving together.

Even in this case the algorithm gives the output in a short time, i.e 102 *ms* for FP-Growth, while running Apriori takes 633 *ms*. This gain of the execution time is very important, especially if you want to apply this method to a longer period of time or to give statistics in near real time.

In addition, when using Apriori to analyse a larger set of paths, some memory errors occurred, both for searching for frequent patterns and for predictions using association rules. When using FP-Growth algorithm, we can generate a sequence of multiple paths, called *association rules*. On the basis of the experiments we deduce that a group will move from point *A* to point *B* and then to other positions, with a certain probability. This is called *confidence* and indicates the conditional probability of being in *B* if you have previously stopped in *A*.

By setting a minimum threshold of 60% we can determine which stations the cyclists will arrive at in the near future. To give an example, the tests show that starting from the time slot 9 – 12 a minimum group of 10 users will move from the “Normandie & Hollywood” bike station to “Figueroa & Cesar Chavez” with Confidence 78%, and the reverse route with Confidence 70%; the group will follow the path “Pacific & North Venice” → “Ocean Front Walk & North Venice” with 61% Confidence; and finally “Ocean Front Walk & North Venice” - “Ocean Front Walk & Navy” with probability 60% (round trip). Figure 7 shows the union of the predicted routes, assuming that the group follows the shortest and most comfortable route to reach all the stages.

For the above prediction experiments, 70% of the available data was used for training. The

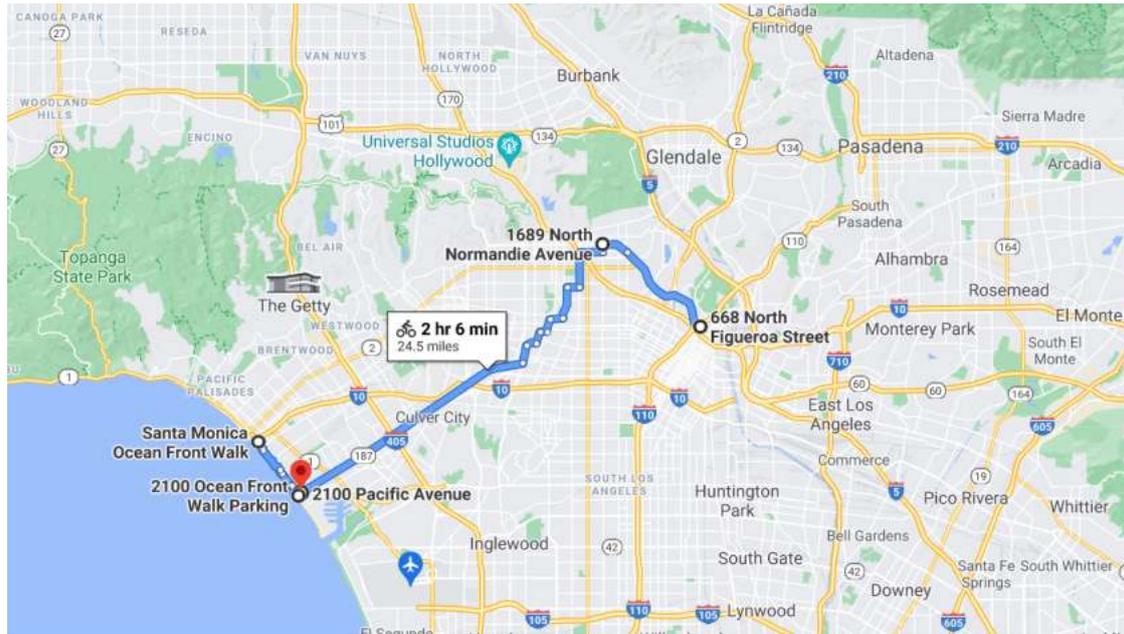


Figure 7: Route prediction for the same group of cyclists. [Google Maps, accessed July, 2021]

remaining 30% of the data was referred to as the validation and test, useful for verifying the results obtained.

7. Concluding Remarks

This paper has shown the feasibility of the proposed approach providing means to enhance a bike sharing service by alerting interested people of bike availability at their preferred bike stations.

Firstly, users are provided with assistant agents as an app on their smartphone that gathers the geographical coordinates when using a shared bike service. Moreover, the app lets users express their preferences when wishing to rent a bike. Secondly, the FP-Growth algorithm has been put at work, and executes on a server to analyse gathered data to provide the said estimates. Such an algorithm has not been used on locations data before.

The experiments have shown that the proposed approach is very fast and then suitable to process the abundance of data available, as it is a linear time algorithm. We have processed a large dataset comprising several thousands of items to recognise common paths of bikes and make estimates of future stops for bikers in a few milliseconds.

Users wishing to use bike sharing services could then use the proposed app to have an estimate of bikes available in a future time slot in a bike station. Then, they could reserve a bike, and be alerted when the bike arrives, or whether the estimated time for availability is updated. Being an estimate availability for the future, though with high probability, sometimes the

availability would be cancelled, however the user would be notified beforehand. We believe that the proposed approach, by letting users plan beforehand their transport means, and schedule, would provide users with a happier experience.

In order to take into account the effects on the use of bikes of the seasonal variability of the weather, as well as changes in people preferences, due to the beginning of the academic year, etc., we consider the most recent data for our statistics, when enough data are gathered to gain sufficient confidence on estimates.

The proposed approach and the implemented solution could be applied to other data related to geographical positions when needing estimates of future locations, rather than bikes. E.g. the movements of electric kick scooters, being very numerous, future drones, and when considering data characterised by GPS positions having fine granularity, even when different kinds of vehicles are considered, could make use of the proposed approach as the linear complexity of the algorithm makes it possible to be applied for a growing amount of data.

8. Acknowledgments

The authors acknowledge project TEAMS funded by University of Catania PIACERI 2020/22.

References

- [1] B. P. L. Lau, M. S. Hasala, V. S. Kadaba, B. Thirunavukarasu, C. Yuen, B. Yuen, R. Nayak, Extracting point of interest and classifying environment for low sampling crowd sensing smartphone sensor data, in: Proceedings of IEEE Pervasive Computing and Communications, 2017.
- [2] M. Aliannejadi, F. Crestani, Personalized context-aware point of interest recommendation, *ACM Transactions on Information Systems (TOIS)* 36 (2018) 45.
- [3] Y.-L. Hsueh, H.-M. Huang, Personalized itinerary recommendation with time constraints using gps datasets, *Knowledge and Information Systems* 60 (2019) 523–544.
- [4] C. Cavallaro, G. Verga, E. Tramontana, O. Muscato, Eliciting cities points of interest from people movements and suggesting effective itineraries, *Intelligenza Artificiale* (2020). doi:10.3233/IA-190040.
- [5] F. de Nijs, G. Theodorou, N. Vlassis, M. M. de Weerd, M. T. J. Spaan, Capacity-aware sequential recommendations, in: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '18, 2018, p. 416–424.
- [6] C. Cavallaro, G. Verga, E. Tramontana, O. Muscato, Suggesting just enough (un)crowded routes and destinations, in: Proceedings of 21st Workshop “From Objects to Agents” (WOA), Bologna, Italy, 14–16 September 2020, volume 2706, 2020, pp. 237–251. URL: <http://ceur-ws.org/Vol-2706/>.
- [7] D. Quercia, R. Schifanella, L. M. Aiello, The shortest path to happiness, in: Proceedings of the 25th ACM conference on Hypertext and social media, ACM, 2014. doi:10.1145/2631775.2631799.
- [8] C. Berzi, A. Gorrini, G. Vizzari, Mining the social media data for a bottom-up evaluation

- of walkability, in: *Proceedings of International Conference on Traffic and Granular Flow*, Springer, 2017, pp. 167–175.
- [9] M. Mythili, A. M. Shanavas, Performance evaluation of apriori and fp-growth algorithms, *International Journal of Computer Applications* 79 (2013).
- [10] C. Borgelt, An implementation of the fp-growth algorithm, in: *Proceedings of 1st international workshop on open source data mining: frequent pattern mining implementations*, 2005, pp. 1–5.
- [11] R. Pensa, A. Monreale, F. Pinelli, D. Pedreschi, Pattern-preserving k-anonymization of sequences and its application to mobility data mining, in: *Proceedings of Privaci in Location-Based Applications (PiLBA)*, volume 397, 2008.
- [12] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu, PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth, in: *Proceedings of 17th International Conference on Data Engineering*, IEEE Comput. Soc, 2004. doi:10.1109/icde.2001.914830.
- [13] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, X. Xu, Dbscan revisited, revisited: why and how you should (still) use dbscan, *ACM Transactions on Database Systems (TODS)* 42 (2017) 19.
- [14] L. Crociani, G. Vizzari, A. Gorrini, S. Bandini, Identification and Characterization of Lanes in Pedestrian Flows Through a Clustering Approach, volume 11298, Springer Verlag, 2018, pp. 71–82. doi:10.1007/978-3-030-03840-3_6.
- [15] B. Ludwig, B. Zenker, J. Schrader, Recommendation of personalized routes with public transport connections, in: *Proceedings of Intelligent Interactive Assistance and Mobile Multimedia Computing*, Springer, 2009, pp. 97–107.
- [16] R. Shourouni, M. Malek, and, Route recommendation based on local users' trajectories, *Journal of Geospatial Information Technology* 4 (2017) 53–67. doi:10.29252/jgit.4.4.53.
- [17] J. M. Kleinberg, Authoritative sources in a hyperlinked environment, *Journal of the ACM* 46 (1999) 604–632. doi:10.1145/324133.324140.
- [18] A. Cecaj, M. Mamei, Investigating economic activity concentration patterns of co-agglomerations through association rule mining, *Journal of Ambient Intelligence and Humanized Computing* 10 (2017) 463–476. doi:10.1007/s12652-017-0665-3.
- [19] J. Heaton, Comparing dataset characteristics that favor the Apriori, Eclat or FP-growth frequent itemset mining algorithms, in: *Proceedings of SoutheastCon*, IEEE, 2016. doi:10.1109/secon.2016.7506659.
- [20] E. Tramontana, G. Verga, Mitigating privacy-related risks for android users, in: *Proceedings of IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2019.
- [21] J. Han, J. Pei, Y. Yin, R. Mao, Mining frequent patterns without candidate generation: A frequent-pattern tree approach, *Data Mining and Knowledge Discovery* 8 (2004) 53–87. doi:10.1023/b:dami.00000005258.31418.83.
- [22] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: *Proceedings of 20th International Conference on Very Large Data Bases, VLDB '94*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994, p. 487–499.
- [23] C. Cavallaro, J. Vitrià, Corridor detection from large GPS trajectories datasets, *Applied*

Sciences 10 (2020) 5003. doi:10.3390/app10145003.

- [24] P. Fournier-Viger, J. C.-W. Lin, B. Vo, T. T. Chi, J. Zhang, H. B. Le, A survey of itemset mining, *WIREs Data Mining and Knowledge Discovery* 7 (2017) e1207. doi:<https://doi.org/10.1002/widm.1207>.

Towards a Semantic Layer for Italian Emergency Plans

Massimo Cossentino¹, Salvatore Lopes¹, Luca Sabatucci¹ and Mario Tripiciano¹

¹CNR (Italian National Research Council), ICAR, Palermo, Via Ugo La Malfa, 153

Abstract

Emergency plans require a complex collaboration among multiple departments and roles. They are generally long textual documents containing practical instructions for hazard responses in natural language. This work focuses on converting informal documents to a more rigorous structured-text representation by taking advantage of well-known techniques from the literature. However, this task is costly, it requires technical skills and sound domain knowledge, and it is entirely subjective. To this aim, we propose a semantic layer that supports the formalization of an emergency plan by identifying essential elements of the input document and highlighting inconsistencies, redundancies, and ambiguities.

Keywords

Emergency Plans, Semantics, Text to Formal Conversion,

1. Introduction

The Italian landscape of emergency plans definition is essentially based on a set of laws issued by the Italian Government that create some milestones for managing emergencies. Firstly, they instituted the Civil Protection agency in 1982. Consequently, they defined the command and control chain, identified responsibilities, and set guidelines for the definition of emergency plans to be written and maintained updated.

Emperor Octavian Augustus, in the first century before Christ, said that “the value of planning diminishes with the complexity of the state of affairs”. In this sentence, he caught the essence of modern planning strategies that relies on *simplicity* and *flexibility*. This approach inspired the purposefully so-called “Augustus Method” [1], introduced in 1997. It addresses a flexible planning approach and defines a simplified way to identify, activate, and coordinate the emergency response assets. The key idea is to overcome the classical approach based on the bureaucratic census of equipment used in civil protection interventions with a new focus on assets availability.

We are considering these laws in the context of N.E.T.TUN.IT¹, a research project for devel-

WOA 2021: Workshop “From Objects to Agents”, September 1–3, 2021, Bologna, Italy

✉ massimo.cossentino@icar.cnr.it (M. Cossentino); salvatore.lopes@icar.cnr.it (S. Lopes);

luca.sabatucci@icar.cnr.it (L. Sabatucci); mario.tripiciano@icar.cnr.it (M. Tripiciano)

🌐 <http://www.pa.icar.cnr.it/cossentino/> (M. Cossentino); <http://www.pa.icar.cnr.it/sabatucci/index.html>

(L. Sabatucci)

🆔 0000-0003-1258-9767 (M. Cossentino); 0000-0003-1488-9755 (S. Lopes); 0000-0003-2852-9355 (L. Sabatucci);

0000-0002-2285-4862 (M. Tripiciano)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

¹Net de l’Environnement Transfrontalier TUNisie-ITalie (N.E.T.TUN.IT), Cross-border Cooperation Programme of the EU Community, Italy-Tunisia 2014-2020

oping a fully operational platform for cross-border data collaboration to cope with shared risks and disasters due to emergency scenarios. In the project, Italian and Tunisian partners simulate the response to an imaginary accident involving atmospheric and marine pollution, nearby population health risks, also considering meteorological issues involved in the scenario.

One of the project's objectives is to support the modelling, verification and run-time adaptation of emergency procedures. The first challenge is that emergency response plans are expressed via informal natural language documents. Converting them into detailed emergency process models is a challenging task, requiring technical skills and deep domain knowledge. Gathering a good understanding of an emergency plan is a costly activity performed through several iterations. Despite all the effort one can make, the same document may rise different interpretations and different conceptual models. For these reasons, we introduce a semantic layer for emergency plans that can facilitate giving a more strict structure to the text-to-formal process.

In this paper, we propose a semantic layer that helps formalise emergency plans with a specific focus on the Italian landscape. Such semantics would help understand the input text, but it would also enable highlighting inconsistencies, redundancies, and ambiguities. In future, we hope to integrate the semantic layer into software platforms for the automatic execution of certain parts of the emergency plan. Anyway, this is the first step for enabling run-time adaptation of the procedures according to contextual events.

The rest of the document is structured as follows: Section 2 introduces the Italian landscape in the definition of emergency plans and it proposes some modelling techniques for providing a more rigorous view of the informal plans. Section 3 provides an overview of the long term agenda on which the current work relies. Section 4 is the core of the paper, presenting the semantics using keywords and templates. Finally, some conclusion is draft in Section 5.

2. State of the Art

Emergency plans use complex protocols, involve collaboration and interaction of multiple departments and roles, containing practical instructions of hazard emergency responses in natural language. This section firstly illustrates the Italian landscape on emergency management, and then it discusses the state of art in modelling them.

2.1. The Italian Landscape

In this section, we will depict the Italian landscape of emergency plan definition. That is based on the so-called "Augustus Method" [1], produced by the Italian Civil Protection Agency and issued by the Ministry of Interior of Italy in 1997.

The Augustus Method sees the territory as a body with organs, called Support Functions, with specific managers that will be made responsible for maintaining alive the plan with drills and updates. In normal conditions, the organs operate according to their duty to maintain the efficiency of the body. When a disease (the emergency) hits the body, all the functions cooperate to heal that.

The Augustus Method defines nine Support Functions for each municipality and five more at the regional level. They include technical advice and planning, health-social assistance

and veterinary services, mass media and information, voluntary organizations, transportation, circulation and traffic, and several other functions.

The Method also clarifies the roles of the different operation centres, from town-level to nation-level granularity. The participation of Support Function managers in these centres ensures the availability of the required assets when needed and their collaboration in facing the emergency.

According to the Augustus Method, the plan is structured in three fundamental parts: (i) General part, (ii) Outlines of Planning, (iii) Model of intervention.

Regarding the General part, it should report all information relating to the knowledge of the territory, the existing monitoring networks, the risks in the area and the related scenarios. The second part (Outlines of the planning) defines the objectives to achieve when an emergency arises. Finally, the Model of intervention assigns responsibilities at the various levels of command and control to manage civil protection emergencies. The constant exchange of information is carried out in the central and peripheral system of the civil protection agencies; resources are listed in the plan and their (continuously updated) capability, location and status of maintenance.

In 2005 a new directive from the President of the Ministry Council refined the structure of the plan. The previous organization deeply inspires the new version, and it includes the following sections: General Part, Incident Scenarios, Organization Model for Intervention, Information to Population, and Cartography.

In this paper we will study 4 plans [2],[3],[4], [5] that have been selected because they represent different examples in terms of scope, size and responsible institution (that authored the plan). They are all reasonably recent (mostly updated in the last five years), and some of them passed through several releases (the most recent is, of course, considered for this study).

We are interested in the active part of the emergency plan, and therefore, we mainly considered the Organization Model for Intervention chapter.

Interestingly, the studied plans show two different outlines for this section: some adopt a 'phase-based' structure, while others a 'role-based' one. The first category (for instance [2]) includes plans for each phase and the role of each stakeholder in that (a chronological description of the tasks and functions by all the stakeholders involved in that phase).

Plans of the second category (for instance [4]) initially describe the different phases (pre-alert, alert, emergency, ...) and which events trigger them. A description of roles and what they should do (one unique list for each position discussing their tasks in all phases) follows. Notably, the second type of structure is preferred in longer documents, likely because it localizes the information regarding each stakeholder, thus facilitating the access to information in an urgency.

This distinction is relevant to this paper because we found that the two structures generate different kinds of ambiguity. Plans in the first category sometimes are vague about who should do a specific task (a consequence of the attempt to avoid repetitions in the text), while plans in the second category eliminate this risk but are less clear in reporting the relationship between each phase and the tasks to be performed within that.

2.2. Modeling Emergency Plans

Emergency plans are usually provided as textual descriptions in natural languages throughout the world. Inevitably, they include ambiguities and imprecise descriptions related to natural

languages. There is a broad research [6, 7, 8, 9] pushing the adoption of (semi)formal representation. The objective is to enable rigorous analysis and validation of process models. Often, a graphical representation provides intuitive means for increasing the understanding of such plans. Notably, the approach we are proposing is intended to represent documents (emergency plans) that are written before the emergency and therefore can be processed offline. Other approaches in the field of emergency management deal with emergency operations enactment the related run-time knowledge [10].

The literature broadly promotes supplementing natural language with standard notations and languages for business processes, such as the Business Process Modeling Notation (BPMN). However, designing high-quality emergency response process models is a great challenge that needs domain knowledge and process modelling techniques. A recent study, mainly focusing on Chinese emergency plans, proposes automatic extraction of business process models from textual descriptions [8]. Another representative example comes from the Norwegian emergency management processes detailed in [9]. The authors claim that authorities and rescuers better understand plans expressed in visual and textual form, and therefore, they can be more proficient in facing unanticipated events. This study also focuses on highlighting roles in the organisations and how they have to interact. The same research recognises some problems using the BPMN standard: some difficulty to model task duration and in reusing a process diagram from one environment to another.

Another study identifies other limitations in the adoption of BPMN for modeling emergency plans [6]. Despite all the advantages, the authors recognise that standard BPMN elements are not comprehensible enough to consider some special requirements of disaster recovery management. For example, the importance of location-related information or multiple resources.

They propose to extend BPMN with some disaster recovery management requirements. In particular, they emphasise the role of resources: the distinction between human and non-human resources, inter-dependencies between resources, various resources properties.

Another branch of research proposes to replace (sometimes to complement) BPMN diagrams with other non-imperative modeling notations. In particular, the Case Management Model and Notation (CMMN) [11] seems a promising candidate.

As stated in the OMG standard², CMMN provides a graphical notation used to capture working methods based on the management of complex cases. CMMN is helpful for those cases that require several activities whose order of execution is unpredictable a priori in response to evolving situations.

Using an event-centred approach and the concept of a case file, CMMN expands the boundaries of what BPMN can model, including less structured work efforts and those driven by knowledge workers. Using a combination of BPMN and CMMN allows for covering a much broader spectrum of work methods.

While traditional business processes can be described by a priori defined sequences of activities using the BPMN notation, the CMMN notation offers more natural support for dynamic workflows. In [12], authors stress the fact that an emergency response is a knowledge-intensive process. To model and automate such process is a challenging task. Authors use CMMN to build a template model for a generic emergency response process.

²<https://www.omg.org/cmmn/>

The OMG also defined a third language, the Decision Model and Notation (DMN), a standard notation for describing and modelling repeatable decisions within organisations. The modelling language triad, BPMN, CMMN and DMN, is defined to cover different areas of process management due to their different focuses. Indeed, in [13], authors investigate how to use a combination of these three modelling languages precisely in the context of crisis management.

3. Motivation: From Free Text Plans to Adaptive WF-based Plans

This section will overview our long-term purpose of introducing adaptive workflows in the execution of an emergency plan.

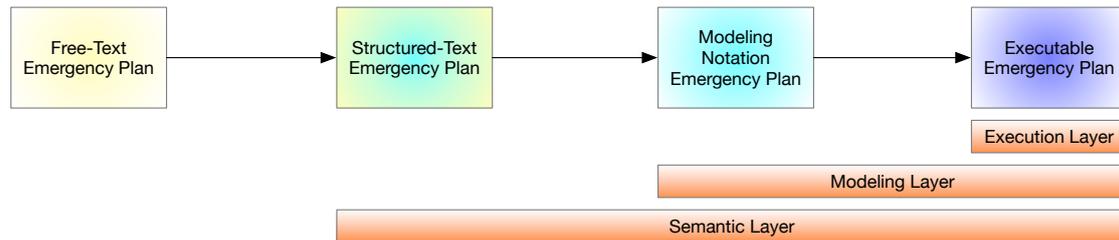


Figure 1: The proposed process for transitioning from free-text emergency plans to executable plans.

As already described, an emergency plan is usually provided in free text form according to a generic format as prescribed by specific laws.

We aim to introduce the support of adaptive workflows in the data exchanges required to face emergencies involving both sides of the Sicilian Channel in N.E.TTUN.IT project. We see that activity as a part of more extensive scope research to introduce adaptation into the execution of the overall emergency plan.

We conceived a process based on the three steps (see also Fig. 1) as described below.

From Free-Text Plan to Structured-Text Plan: this is the step described in this paper. It includes two sub-steps: (i) the construction of a set of keywords used to (ii) fragment the free-text plan in a list of notable items for its representation in a specifically conceived modeling notation (see below). The main contribution of this paper is the semantic layer for processing the free-text plan and producing a structured-text version (see Section 4). This layer includes a list of keywords we propose to use to represent a structured-text version of the informal plan. The idea is that these keywords will be specifically supported by the proposed modeling notation and will receive a specific implementation in the execution layer.

Representing the Structured-Text Plan with a Modeling Notation: we are devising a specific modeling notation to represent an emergency plan. An essential requirement for this notation is that it will easily support adopting an adaptive middleware for the execution and coordination of the plan's activities. The definition of this notation is still a work-in-progress activity, but we have identified the main contributions it will receive from a few well-known standards. The BPMN notation is a part of that but not so central as it could be expected. The

Augustus directive prescribes that a plan provides general indications for the management of the emergency, whereas details are to be defined at emergency time. For this reason, we are considering to adopt the Case Management Modeling Notation (CMMN) that allows to represent scenario-based situations, and the Decision Modeling Notation (DMN) that allows to formalise critical aspects of decisions to be taken during the development of an accident, also including decision criteria (like data values reported by personnel on scene) and the reference documents to be consulted. Finally, some parts of the free-text plans naturally convey the opportunity to introduce a model of the goals related to the responsibilities of involved stakeholders (like the authorities and the support functions described in a plan).

Adaptive Execution and Management of Emergency Plan: this part of the process presents some interesting challenges: first of all, an essential issue of emergency management is in the central role of the persons in charge at the different levels of the command chain. They have to maintain complete control of the running activities, and they need updated information, like the availability of resources, for proper accident response.

Unexpected events may spoil the validity of any detailed a-priori planning: the actual plan has to emerge from the consideration of the specific accident, environmental conditions (for instance roads practicability, weather, and so on), assets (fuel, food, transportation, and so on), operational capability (for instance their numerical consistency together with their intervention time and need for logistic support). These considerations raise interesting issues from the scientific point of view, since they prescribe the availability of an accident management system that includes relevant features of adaptive workflows where human resources play a key role not only as activity performers (human-intensive system) but also in the decision making role. Stakeholders involved in playing the decision role need to be made well-aware of the situation. Therefore they need specific support to access just-in-time precise and updated knowledge. Decision support systems may help in the decision phase, also referring to archive data and simulated scenarios. The system also needs a spatial-referred representation of the environment and involved assets. Finally, communication capability is a key element. The support for emergency communications is already existing, and great care is devoted to the adoption of standardized content protocols for messages, as the Common Alerting Protocol (CAP) [14] that is an XML-based data format for exchanging public warnings and emergencies. Although that is of relevant value, more is still to be done on the telecommunication infrastructure resilience and the automatic support for alternative delivery channels.

4. The Proposed Semantic Layer for Emergency Plans

This section describes our work on the semantic layer that we use to process the free-text plan and produce the structured-text plan manually. This layer blends lexical semantics (i.e. investigates word meaning) and conceptual semantics (explaining properties of argument structure). In other words, we propose a set of keywords to be either extracted from the input text or derived from the context. A keyword is an (Italian or English) word that has a specific meaning for modelling purposes. Each keyword has a precise definition, and it is related to a set of properties.

It is now necessary to clarify some rules we adopted in defining the keywords:

- Keywords must belong to the general structure of the emergency response plan: this means that domain actions related to the management of a specific accident (even if common to other cases) are not part of the semantics. The reason for this choice is to limit the number of keywords and to remain general by leaving apart domain-dependent items that change with the kind of emergency or that may depend on the adoption of new strategies and new technologies.
- When possible, keywords (or their synonymous) should appear in the free-text version of the plan. Our idea is to respect the problem knowledge and comprehension that the writer of the plan has. Some keywords expressing decisions and sometimes goals are not explicitly named in the plan's text, but the document's structure helps deduce them.

Each keyword is therefore described by a *label* (the keyword itself), a *list of synonymous* (addressing the same meaning in the plan and sometimes used to avoid repetitions), the *addressed meaning* (a clarification of the meaning to improve the comprehension of the way the keyword is to be intended, also with the use of examples), a *sentence template* (used to exemplify the common use of the keyword), and a *table of properties* (like the actor responsible for performing an action or the target of a message).

We provide the table of properties with a twofold objective: (i) it helps to clarify the syntactic role of words in the free-text version of the plan; (ii) it helps in disambiguating situations in which the same word fits different keywords (with slightly different meanings).

For instance, a Message Event is always produced by a source and consumed by a target. This essential feature is rendered by the 'from' and 'to' properties.

Because of their object-oriented nature, in this work, tables of properties may be related one to another by inheritance relationships. The motivation is basically for clarity and saving space: notably, as we will report later on, some properties are common to many keywords. Some elements are mandatory in a table of properties for assigning a meaning to the corresponding keyword, whereas others are optional. Graphically, we used an asterisk for intending required elements.

The designer will parse the free-text plan to identify the keywords (or their synonyms). For each identified keyword, she will fill the corresponding table of properties. When the table of properties is complete, then the designer may instantiate the sentence template. This is a syntactic structure aiding to recognise words with given meaning in sentences. In other words, it represents a shortcut for the keyword in the specific context to be used for modelling purposes. For instance, it can represent the name of a BPMN task.

The rest of this section discusses keywords and some of their most relevant properties deduced from the free-text plan.

As already discussed, the proposed semantics will represent the emergency plan except the domain-specific actions used to solve the accident.

We group the resulting keywords into five categories: actors, events, actions, decisions and responsibilities (goals). We detail these categories in the following subsections.

Although these categories emerge from the list of empirically discovered keywords found by processing the plans, we should also consider that they relate somehow to the parallel definition

of a modelling notation we are performing at the same time.

While actors and actions have evident relevance in a plan, events have a motivation. They trigger changes in the different phases of the plan (for instance, from pre-alert to alert) or trigger responses to the accident's evolution reported by observations or incoming news. Responsibilities represent why a specific actor is involved in the emergency management plan, what she should care and pursue, what she could be considered accountable for according to the law. We devote specific attention to the decision category for the impact decisions have on the execution of the overall plan; more on that in subsect. 4.4 .

4.1. Actors

A fundamental aspect of the innovation proposed by the Augustus method consists in clearly assigning responsibilities. For this reason, it becomes very relevant to create a list of actors involved in the execution of the plan. The list includes names, acronyms and the definition of the actor.

In the numerous plans we studied, we found many actors, often referred to with acronyms. We noted that the ambiguity allowed by the Italian language sometimes creates indecision about the actor responsible for performing a specific action. This indecision mainly happens in plans where the description is ordered using time or event-related criteria. Plans, where activities are clustered according to actors, do not present this ambiguity, of course. The list of actors is long and includes authorities (Mayor, Prefect, ...), personnel with specific skills (engineers), and groups (committees, operation/command centres, associations of volunteers, ...). We will not entirely report them for the sake of conciseness and because the complete list would add nothing significant to the proposed argumentation.

We differentiate between individual actors and collective actors. With the term individual actor, we will address the common-sense meaning of a participant in an action or process. Collective actors represent a more refined concept where according to [15] collective actors perform a coordinated and collaborative decision-making process where one individual speaks for the group. Collective actors share the same interests, integration mechanisms, an internal and external representation of the collective actor and an innovation capacity.

Examples of individual actors include some of the already cited authorities: Mayor, Prefect, chairs and participants of committees (that are collective actors), for instance, the Responsible of the Town Operating Centre or the Civil Protection Officer on duty.

Examples of collective actors include the operation room of the Metropolitan Police, the Rescue Coordination Center, the Regional Agency for Environment Protection, the Integrated Regional Operation Room, and so on.

4.2. Events

Emergencies exist because events arise in the environment; this is a simple fact that justifies the importance of events in the field.

Emergencies exist because events arise in the environment; this is a simple fact that justifies the importance of events in the field.

From our analysis, we recognised two types of relevant events for the structured representation of the plan: Data Events and Messages.

Data incoming represents the arrival of some data from monitoring activities or the acquisition of information from any possible source. The Message event represents the arrival of a message related to the emergency, for instance, a phone call from the responsible manager of a plant affected by a significant fire blast. Messages belong to two subcategories: Informal and Formal. Informal messages refer to phone calls, media diffusion of news, and so on. The essential feature of an informal message is that it does not have any kind of template specified in the emergency plan. Formal messages are delivered using traceable communication means (emails, other types of computer-based messages, telegrams,...). An essential feature of formal messages is the adherence to a format specified in the plan. Frequently formal messages are encoded using some emergency communication protocol, like the Common Alert Protocol (CAP) [14].

It is notable to consider that incoming/outgoing data are different from messages since with the data event, we represent the availability of the information, for instance, the read from a sensor or the output of a simulation. Of course, data events reach some actors using a message whose arrival/departure are other events occurring at successive times.

In the structured-text plan, we represent events extracted from the free-text plan by using the sentence templates proposed in the following sections, together with the list of their properties. Notably, all events share one specific property, the timestamp, since the concept of the event has a well-defined location position in time.

4.2.1. Data Event

As regards the *Data Event*, the template is:

Data Event [source/target] [thresholds] [values]

The 'source/target' represents the originator of the data (for input data) or the recipient (for output data), for instance, a sensor or a computer application. 'Thresholds' specifies the thresholds that are to be monitored, for instance, the level of a river that should not reach one meter. The reading of such level measure is the data event that has some significance for accident management; 'values' represent the information enclosed in the event; it may include the unit of measure or other significant information.

The presence of the 'threshold' property may seem odd in this event type since it is necessary to perform some action to compare the actual reading of a sensor with the threshold value. Although formally speaking, this is true, in the plan, we often found that data readings become significant only when they reach or overcome some specific value (for instance, a pollutant concentration in the atmosphere). All data readings that do not reach the threshold are not significant, and thus they do not generate a 'Data' event (if a threshold is specified).

4.2.2. Message Event

The *Message Event* is detailed by using the following template:

[Formal/Informal] Message [subject]

'Formal/Informal' reports the adoption, for the message, of a well-defined syntactic structure, often included in in the free-text plan, the 'subject' field contains a short description of the message content, 'properties' include a list of slots as discussed in the table below:

'Message' properties	
From *	Sender actor
To *	Recipient actor or actors' list
Priority	Priority of the message
Content*	The information content of the message
Communication channel	The channel used to dispatch the message
Template	Reference to the message template from the plan
Exception	Description of how to deal with unsuccessful delivery of the message

Significantly, in some plans, we found the explicit indication of how to deal with the impossibility to reach the destination person (for instance, by contacting a specified alternate contact person). For this reason, we introduced the Exception slot in the properties list.

4.3. Actions

Actions are the essential brick of an emergency plan. Actions can be atomic or composed; initially, we will refer to an abstract concept of 'action' that will found a realisation in a set of concrete keywords of the semantic layer: orders, activates, arranges, gathers data, informs, plus generic domain-specific acts.

All the keyword's tables of properties inherit the properties of a mother class *Action Properties*. This factored all the common properties the other templates inherit and specialize, and allows not to repeat them in every table:

'Action' properties	
owner (actor)*	is responsible for the action and operates for its enactment
on-behalf-of (actor)	<i>delegates the action</i>
assistant (actors)	<i>supports the owner's duty in the enactment</i>
precondition	<i>execution context</i>
temporal constraints	<i>start, duration, period, and due time</i>
resources	<i>(min/max) assigned resources</i>
quality requirements	<i>minimal expected quality of service</i>
ack to	<i>when present, a notification is required when completed</i>

In the following, we will avoid repeating the fields common to all templates. We will mention only cases in which there is a clear difference of meaning.

The first keyword we specialise from *Action* is suitable for situations in which an authority sets an order via either a formal or informal message. This action is typically described by a template sentence like:

[authority] Orders [personnel] to [do something/address an outcome]

When the act is formal, the order is issued by some decree, ordinance,..., sometimes the order does not concern a specific person, but it is rather directed to all the citizens. An example occurs when the Civil Protection Office Head in a town (after having been delegated by the Mayor to do that) orders to close some roads to the Metropolitan Police Chief. The properties of the 'Orders' keyword are reported below:

Orders	
Matching with Italian words: ordina, disporre	
owner (actor)*	issues the order
on-behalf-of (actor)	<i>delegates the order</i>
target (actor)*	receives (executes) the order
object*	the action/goal to be undertaken
formal deed	<i>legal/official document issuing the order</i>
comm channel	<i>how the order is communicated to the target actor</i>

Another keyword that we specialise from *Action* is *Activates*. It covers situations where a state transition is required. It is similar to the keyword 'orders' but describes an operation and could be the consequence of the order. The corresponding sentence is:

[somebody] Activates [something]

Differently from *Orders*, here the personnel is an actor responsible for enabling something (a sub-plan, an office, a function) to become working/operative.

Examples:

- the Civil Protection Office Head activates the weather monitoring team;
- the emergency manager activates the External Emergency Response Plan.

Activates	
Matching with Italian words: attiva, pone in essere	
owner (actor)*	is responsible for the transition
object/state*	the target to be activated

We also considered the *Arranges* keyword. It resembles the *Orders* and *Activates* ones but it implies some kind of preliminary coordination/organization/planning before the subject becomes active. This is a frequent occurrence in the plans we studied. The sentence template is the following:

[someone] Arranges [a subject]

The properties list is reported below:

Arranges	
Matching with Italian words: predisporre, prepara	
owner (actor)*	is responsible of planning/organising/...
participant actors	<i>supports the target actor(s) in performing what planned/organised by the owner actor</i>
object*	the plan/organization to be enacted

Very central in dynamic contexts, such as those of emergency responses, the *Gathers Data* keyword refers to the act of acquiring data from direct observation, from monitoring sensors, from experts on the field, and, sometimes, from citizens. It is also used for checking the state of resources. The corresponding sentence template is:

[someone] Gathers Data about [a subject]

where the subject specifies the kind of data to be acquired/monitored.

Gathers Data	
Matching with Italian words: raccoglie informazioni, opera monitoraggio	
owner (actor)*	is responsible of collecting data
assistant actors	<i>aid in collecting data</i>
subject*	data, information or event to be acquired/monitored
data source	<i>the source of the data</i>
frequency	<i>specifies the frequency of data acquisition</i>

Examples:

- The Regional Civil Protection Agency gathers information about the accident.
- The head of the Resources Department gathers data about the status of resources and personnel.
- The head of the Civil Protection Agency identifies suitable places to shelter displaced persons.

The last example shows the use of a synonym ('identifies' in place of 'gathers data') in the proposed sentence. The use of synonymous is quite common, especially in the Italian language, and it will be discussed in Subsect. ??.

This sentence is also an intriguing case of ambiguity. It could also be intended as a decision about what are the best places to shelter displaced persons. However, we deduced that the sentence refers to a 'gathers data' act by looking at the context.

Informs is the last non-domain-dependent keyword, derived from Action. It represents the act of sending formal/informal messages. It concerns situations in which:

[somebody] Informs [recipients] about [an object]

where the message may be a formal act, following a structured protocol, or an informal communication like a phone call. The action 'Informs' may be done once or repeated every time an event occurs (thus updating the information).

For example: the head of the Civil Protection office informs the commander of the Municipal Police about the emergency status.

Informs	
Matching with Italian words: informa, comunica, aggiorna, mantiene informato	
owner actor*	is responsible of outgoing messages
<i>assistant actors</i>	<i>aid in sending messages</i>
object*	sent data, information, or acknowledgement
recipient actors*	will receive the information
<i>regularly</i>	<i>when flagged, the activity is repeated every time there are updates</i>
<i>formal deed</i>	legal/official document issuing the order to inform
means	how the order to inform is notified to the owner actor that will execute it

As a prototype sentence for the final item of the semantic layer, we propose an abstract *Acts* keyword that is to be instantiated in the different domain-specific actions performed to face a disaster:

[somebody] [Acts] [on something]

The *Acts* keyword is the mother class of a range of domain-based actions that are not classifiable in the previous categories. One or more actors can operate it as the consequence of an order or a plan, an event may also trigger it. An action could not be allowed (for instance for security reasons) in some phases of the emergency.

Examples are: (i) on-site paramedics provide first treatment and help in evacuation of injured people, (ii) bulldozers remove debris from the road, (iii) the Metropolitan Police deploys road signs to prevent unauthorised access.

Acts	
Matches with Italian words: (related to the domain)	
owner (actor)*	is responsible of action execution
input	<i>expected input data necessary for action completion</i>
post-condition	<i>expected outcome of the action</i>

4.4. Decisions

Decisions are a type of action requiring information, quick access to knowledge (i.e. maps, technical schematics of plants) or expertise (i.e. support from technical staff). Decisions in the plans are often described in terms of the actor who has to take them and the possible alternatives (i.e. trigger alert or deactivate the pre-alert phase). Sometimes some supporting actors are also

listed. Decisions are the part of the plan that we have often found lacking relevant details; for instance, criteria for deciding may remain blurred and rarely formalised.

Often, plans do not explicitly use the 'decides' word; they instead address the concept of a decision to be taken by someone by describing the incoming events and expected decisions in terms of orders issued or actions undertaken. Sometimes, events may be related to the emergency development and decisions regarding the acts necessary to perform to face the new event.

In order to define a template for the decision act, we use the following sentence:

[somebody] Decides [something]

where 'somebody' is the main responsible for the decision, and can be supported by assistants.

The placeholder 'something' concerns the outcome of the decision process. Typically taking decisions requires specific background knowledge and contextual data. Moreover, norms, criteria, and best practices that are prescribed to respect during the emergency development.

This keyword may not be found in the emergency plan; the interpretation of a portion of text rather generates an instance of the sentence template that we will introduce in the structured version of the plan to represent the act of deciding something as a consequence of some data/event

Decides	
Matching with Italian words: decide, stabilisce, assimila, delibera	
owner (actor)*	gets the responsibility for the decision
on-behalf-of (actor)	<i>has delegated the decision</i>
assistant actors	<i>aid in taking the decision</i>
temporal constraints	<i>start, duration or due time</i>
input	<i>expected input data/event, necessary for taking the decision</i>
knowledge	<i>required background information for taking the decision</i>
criteria	<i>formalized rules as a support for the decision</i>
output*	<i>the outcome of the decision</i>

4.5. Goals

As issued by Italian law, an emergency plan assigns specific responsibilities to the participant actors. For instance, according to the Augustus Method, each Support Function manager is in charge of a specific responsibility such as ensuring health-social assistance, managing mass media and information, organizing voluntary organizations, controlling circulation and traffic and so on. Responsibility is the commitment to address an objective under the criterion of personal responsibility that means the actor can be prosecuted if results are not expected.

We labelled this keyword as *Goal* because we intend to create a correspondence with strategic actor relationships that originates from social modeling [16] and Goal-Oriented requirements engineering [17] and to pave the way for multi-agent system automatic support. In this perspective, responsibilities (hereafter we refer to them as goals) lead to direct actions or delegations to other parties. A template sentence for the goal is:

[precondition], [someone] is responsible for [objective]

Properties of the goal keyword are reported below:

Goal	
owner (actor)*	is responsible of pursuing the goal
objective*	the objective that the owner actor will pursue
alternate owner (actor)	is responsible when the owner is not available or reachable
on-behalf-of	<i>the owner acts on behalf of this collective actor</i>
participant actors	<i>support the owner actor in performing her duty</i>
precondition	<i>preconditions to be verified for triggering goal pursuing</i>

Several parts of the emergency plan assign responsibilities to actors involved in emergency management. However, as well as happened for decisions, goals are often implicitly defined in the emergency plan. Indeed, the primary goals can be extracted by the Support Functions that are explicitly listed in the plan as the Augustus Method prescribes. Other goals require additional effort in the identification (task supported by using the property table). In the following, we will present some examples of goals extracted from the analysed plans:

- during the Alarm Phase, the Chief of the Provincial Fire Brigade is responsible for coordinating the technical and scientific staff;
- the Provincial Health Agency General Manager is responsible for activating the necessary organisation for the specific type of accident;
- the Chief of the City Brigade Fire is responsible for coordinating all operative structures forming the Rescue Coordination Centre.

4.6. Synonyms

To write the present work we have studied and analysed a number of Italian Emergency Plans [2],[3],[4], [5] to compare the different approaches, styles and terminologies.

Each author expresses the guidelines using her sensibility and linguistic knowledge; this often leads to using different words (synonyms) to address the same event or the same action. The use of synonyms may even occur in the same document where, for the sake of elegance and avoiding repetitions, the author uses synonyms to address the same concept.

To identify these linguistically different but semantically similar words, we have defined a correspondence table. A portion of this table is shown in Table 1. A synonymous table is significant because it limits the arbitrariness in interpreting the free-text plan and passing from it to the structured-text version of the same plan.

Table 1
Table of Synonyms

Term from the Emergency Plan (Italian)	Corresponding Term in English	Synonymous Keyword in our semantics	Example
Dispone	Arranges	Orders	The prefect arranges a continuous monitoring of air quality in relation to wind direction, intensity and height
Aggiorna	Updates	Gathers Data	The Manager shall update the Prefect and other interested parties by means of messages using the template specified in the attachments
Segue l'evoluzione	Monitors	Gathers Data	The Mayor monitors the development of the situation and informs the population that the state of emergency has been revoked.
Acquisisce	Acquires	Gathers Data	The operations room manager acquires all relevant updates on the status of technical and rescue operations.
...			
Istituisce	Sets up	Activates	The Technical Rescue Director sets up an Advanced Command Post using his own vehicle.
Pone in essere	Implements	Activates	The Manager implements the Internal Emergency Plan

5. Conclusions and Future Works

This paper focused on inconsistencies, redundancies, and ambiguities that hinder understanding and formalizing emergency plans. The need to convert informal documents to more rigorous conceptual models require a semantic layer for identifying essential elements of the input text and resolving linguistic issues that may be present. We extracted essential keywords through an empirical study of several Italian documents reporting different kinds of emergency plans. Our analysis allowed us to discover recurrent structures in these documents. Sometimes, these linguistic structures are evident, other times are hidden, and some interpretation of the text's meaning is needed. We support identifying them by using templates with required and optional properties to be filled in all the cases. We will use the proposed approach in the context of the N.E.TTUN.IT research project. We plan to provide automatic support for the execution of emergency procedure plans, with the long-term objective of enabling the properties of robustness and adaptivity.

6. Acknowledgments

The research was conducted for the Net de l'Environnement Transfrontalier TUNisie-ITalie (N.E.T.TUN.IT) research project, founded by Cross-border Cooperation Programme of the EU Community, Italy-Tunisia 2014-2020.

References

- [1] E. Galanti, Il metodo augustus, DPC Informa "Periodico informativo del Dipartimento della Protezione Civile" (1997).
- [2] P. B. F. Mereu, A. Alferi, Piano di protezione civile per la salvaguardia della popolazione dal rischio idrogeologico derivante dalla ostruzione del canale sotterraneo papireto ed 2015-16, Technical Report, Comune di Palermo, 2015.
- [3] Protezione Civile Provincia di Ravenna, Piano Provinciale di Emergenza Piano Provinciale di Emergenza Rischio Chimico-Industriale, Technical Report, Prefettura di Ravenna - Provincia di Ravenna, 2009.
- [4] Prefettura di Siracusa, Pianificazione di emergenza esterna per gli stabilimenti a rischio di incidente rilevante, Technical Report, Siracusa, 2019.
- [5] Prefettura di Palermo, Piano Emergenza Esterna Esso Italiana S.r.l Deposito oli minerali Via Messina Marine 813 Palermo, Technical Report, Palermo, 2016.
- [6] H. Betke, M. Seifert, Bpmn for disaster response processes, INFORMATIK 2017 (2017).
- [7] N. T. T. Le, C. Hanachi, S. Stinckwich, T. V. Ho, Mapping bpmn processes to organization centered multi-agent systems to help assess crisis models, in: Computational Collective Intelligence, Springer, 2015, pp. 77–88.
- [8] W. Guo, Q. Zeng, H. Duan, G. Yuan, W. Ni, C. Liu, Automatic extraction of emergency response process models from chinese plans, IEEE Access 6 (2018) 74104–74119.
- [9] V. Nunavath, A. Prinz, Norwegian emergency management process by using business process modeling notation, in: Proceedings of the 8th IADIS International Conference on Information Systems 2015, 2015.
- [10] P. Brunetti, A. Croatti, A. Ricci, M. Viroli, Smart augmented fields for emergency operations, in: Proc. of The 5th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2015), 2015.
- [11] M. A. Marin, Introduction to the case management model and notation (cmmn), arXiv preprint arXiv:1608.05011 (2016).
- [12] A. Y. Shahrah, M. A. Al-Mashari, Modelling emergency response process using case management model and notation, IET Software 11 (2017) 301–308.
- [13] M. P. R. Herrera, J. S. Díaz, Improving emergency response through business process, case management, and decision models., in: ISCRAM, 2019.
- [14] OASIS Standard, Common Alerting Protocol Version 1.2, <http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html>, 2010. Accessed: 01 July 2010.
- [15] A. Pichierri, Lo sviluppo locale in Europa. Stato dell'arte e prospettive, Rubbettino, 2005.
- [16] S. Eric, P. Giorgini, N. Maiden, J. Mylopoulos, Social modeling for requirements engineering, MIT press, 2011.

- [17] A. Van Lamsweerde, Goal-oriented requirements engineering: A guided tour, in: Proceedings fifth iee international symposium on requirements engineering, IEEE, 2001, pp. 249–262.

Towards cooperative argumentation for MAS: An actor-based approach

Giuseppe Pisano¹, Roberta Calegari¹ and Andrea Omicini²

¹*Alma AI – Alma Mater Research Institute for Human-Centered Artificial Intelligence, ALMA MATER STUDIORUM—Università di Bologna, Italy*

²*Dipartimento di Informatica – Scienza e Ingegneria (DISI), ALMA MATER STUDIORUM—Università di Bologna, Italy*

Abstract

We discuss the problem of cooperative argumentation in multi-agent systems, focusing on the computational model. An actor-based model is proposed as a first step towards cooperative argumentation in multi-agent systems to tackle distribution issues—illustrating a preliminary fully-distributed version of the argumentation process completely based on message passing.

Keywords

Argumentation, MAS, cooperative argumentation, distributed argumentation process

1. Introduction

One of the most critical problems in distributed and collaborative multi-agent systems (MAS) – where agents cooperate towards a goal – is conflict resolution, where *argument evaluation* often plays a critical role [1]: agents can provide explicit arguments or justifications for their proposals for resolving conflicts by exploiting the so-called negotiation via argumentation, or *cooperative argumentation*, as an effective approach to resolving conflicts. There, the purpose of multi-agent argumentative dialogues is to let agents reach an agreement on (i) the evaluation of goals and corresponding actions (or plans); and (ii) the adoption of a decentralised strategy for reaching a goal, by allowing agents to refine or revise other agents' goals and defend one's proposals.

Cooperative argumentation is exploited in some real-world multi-agent applications [2]. However, a key problem in such applications is that a widely-acknowledged well-founded computational model of argumentation is currently missing, thus making it difficult to investigate the convergence and scalability of argumentation techniques in highly-distributed environments [1, 2]. To alleviate those difficulties, we present a first version of a message-based distributed argumentation algorithm as the basic pillar of a computational model for cooperative argumentation in MAS. In this work we ignore issues such as agent autonomy and MAS coordination artefacts, and focus instead on the distribution issues of cooperative argumentation, based on the logic-based agreement framework Arg2P [3, 4], which enables agent dialogue and defeasible reasoning in MAS. In particular, we focus on the single-query evaluation mode

WOA'21: 22nd Workshop "From Objects to Agents", September 01–03, 2021, Bologna, Italy

✉ g.pisano@unibo.it (G. Pisano); roberta.calegari@unibo.it (R. Calegari); andrea.omicini@unibo.it (A. Omicini)

🆔 0000-0003-0230-8212 (G. Pisano); 0000-0003-3794-2942 (R. Calegari); 0000-0002-6655-3869 (A. Omicini)



© 2020 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

of the tool, aimed at evaluating the admissibility of a single statement with no need to build the entire argumentation graph. We propose a preliminary fully-distributed version of the argumentation algorithm, based on message passing, whose focus is on the requirements for a sound distributed evaluation of the argumentation task. For the purpose of this paper we exploit the actors' paradigm and its main properties—i.e., (i) fully-reactive computational nodes, and (ii) communication through message passing.

Accordingly, the paper is structured as follows. In Section 2 we give an overview of the main themes on which the paper is focused. Section 3 and Section 4 illustrate the contribution, introducing a distributed computational model that enables the assessment via argumentation of a single argument. In particular, Section 3 first discusses how the argument evaluation algorithm of Arg2P can be parallelised, then addresses the problem of knowledge manipulation in a decentralised setting. In Section 4 we deliver a complete and coherent model for decentralised reasoning based on the actor model. Finally, in Section 5 we provide the final remarks and discuss the aspects of this work that are still open for improvement and future research.

2. Preliminaries

2.1. A basic intro to structured argumentation

In the argumentation language, a literal is an atomic proposition or its negation.

Notation 1. For any literal ϕ , its complement is denoted by $\bar{\phi}$. That is, if ϕ is a proposition p , then $\bar{\phi} = \neg p$, while if ϕ is $\neg p$, then $\bar{\phi}$ is p .

Literals are brought into relation through rules.

Definition 1 (Rules). A *defeasible rule* r has the form: $\rho : \phi_1, \dots, \phi_n \Rightarrow \psi$ with $0 \leq n$, and where

- ρ is the unique identifier for r ;
- each $\phi_1, \dots, \phi_n, \psi$ is a literal;
- the set $\{\phi_1, \dots, \phi_n\}$ is denoted by $Antecedent(r)$ and ψ by $Consequent(r)$.

Defeasible rules – denoted with $DefRules$ – are rules that can be defeated by contrary evidence. Pragmatically, a defeasible rule is used to represent defeasible knowledge, i.e., tentative information that may be used if nothing could be posed against it. For the sake of simplicity, we define non-axiom premises via defeasible rules with empty *Antecedent*. A theory consists of a set of rules.

Definition 2 (Theory). A *defeasible theory* is a set $Rules \subseteq DefRules$.

Arguments are built from defeasible rules. Given a defeasible theory, arguments can be constructed by chaining rules from the theory, as specified in the definition below—cf. [5].

Definition 3 (Argument). An *argument* A constructed from a defeasible theory $\langle Rules \rangle$ is a finite construct of the form:

$$A : A_1, \dots, A_n \Rightarrow_r \phi$$

with $0 \leq n$, where

- r is the top rule of A , denoted by $\text{TopRule}(A)$;
- A is the argument's unique identifier;
- $\text{Sub}(A)$ denotes the entire set of subarguments of A , i.e., $\text{Sub}(A) = \text{Sub}(A_1) \cup \dots \cup \text{Sub}(A_n) \cup \{A\}$;
- ϕ is the conclusion of the argument, denoted by $\text{Conc}(A)$;

Arguments can be in conflict, accordingly to two kinds of attack: rebuts and undercutting, here defined as in [5].

Definition 4 (Attack). An argument A **attacks** an argument B (i.e., A is an **attacker** of B) at $B' \in \text{Sub}(B)$ iff A undercuts or rebuts B (at B'), where:

- A undercuts B (at B') iff $\text{Conc}(A) = \overline{\text{TopRule}(B')}$
- A rebuts B (at B') iff $\text{Conc}(A) = \bar{\phi}$ and $\text{Conc}(B') = \phi$

An argumentation graph can then be defined exploiting arguments and attacks.

Definition 5 (Argumentation graph). An **argumentation graph** constructed from a defeasible theory T is a tuple $\langle \mathcal{A}, \rightsquigarrow \rangle$, where \mathcal{A} is the set of all arguments constructed from T , and \rightsquigarrow is the attack relation over \mathcal{A} .

Notation 2. Given an argumentation graph $G = \langle \mathcal{A}, \rightsquigarrow \rangle$, we write \mathcal{A}_G , and \rightsquigarrow_G to denote the graph's arguments and attacks respectively.

Given an argumentation graph, we leverage on labelling semantics [6, 7] to compute the sets of arguments that are accepted or rejected. Accordingly, each argument is associated with one label which is either IN, OUT, or UND—respectively meaning that the argument is either accepted, rejected, or undecided.

2.2. Structured evaluation in Arg2P

The Arg-tuProlog (Arg2P in short) [3, 4] engine is a logic-based agreement framework enabling defeasible reasoning and agents' conversation, which reifies the structured argumentation model presented above.

With respect to the available argumentation frameworks, Arg2P includes the *query-based* mode, which allows for single-query evaluation according to the selected semantics¹. Single-query evaluation is precisely the algorithm we are interested in, given that cooperative argumentation in highly-reactive systems is often based on a quick debate on some beliefs – those

¹At the time of writing, only grounded semantic is fully implemented

Listing 1: Structured argumentation, Arg2P answer query algorithm for grounded semantic (pseudo-code).

```

AnswerQuery(Goal) :
  A1, ..., An = buildSustainingArguments(Goal)
  Res = ∅
  for A in A1, ..., An:
    Res = Res ∪ Evaluate(A, ∅)
  return Res.

Evaluate(A, Chain) :
  if(∃ B ∈ Attacker(A): Evaluate(B, A ∪ Chain) = IN)
    return OUT
  if(∃ B ∈ Attacker(A): B ∈ Chain)
    return UND
  if(∃ B ∈ Attacker(A): Evaluate(B, A ∪ Chain) = UND)
    return UND
  return IN.

```

concerning the decision to be made at that moment – rather than on a complete assessment of all the agents’ knowledge—where a shared agreement is not easily achieved.

This feature is accessible in the tool through the predicate

$$\text{answerQuery}(+Goal, -Yes, -No, -Und)$$

which requests the evaluation of the given *Goal*, and gets the set of facts matching the goal distributed in the three sets IN, OUT, and UND as a result.

The algorithm used to evaluate a single claim (or query) according to grounded semantic is inspired by the *DeLP* dialectical trees evaluation [8]. Listing 1 shows the pseudo-code – AnswerQuery(Goal) – for the answerQuery/4 predicate: given a claim (Goal) as input, the function first builds all the arguments sustaining that claim (buildSustainingArguments(Goal)), and then requires their evaluation via the Evaluate(A, Chain) function. To assess the A_1, \dots, A_n status (acceptability or rejection), three conditions are evaluated—whose order is important to ensure the soundness of the algorithm:

- (Cond1) if a conflicting argument labelled as IN exists, then A_1 is OUT;
- (Cond2) if a cycle in the route from the root to the leaves (Chain) exists, then A_1 argument is UND;
- (Cond3) if a conflicting argument labelled as UND exists, then also the A_1 argument is UND.

If none of the above conditions is met then the argument can be accepted.

Example 1. Let us consider the following theory and the corresponding arguments (also depicted in Figure 1)

$r1 : \Rightarrow a$	$A0 : \Rightarrow_{r1} a$
$r2 : a \Rightarrow b$	$A1 : A0 \Rightarrow_{r2} b$
$r3 : \Rightarrow \neg b$	$A2 : \Rightarrow_{r3} \neg b$
$r4 : b \Rightarrow c$	$A3 : A1 \Rightarrow_{r4} c$

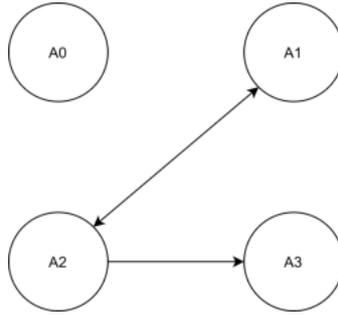


Figure 1: Argumentation graph for arguments from Example 1, in which nodes are arguments and edges are attacks between arguments.

where, according to grounded semantics $A0$ is IN – there are no arguments contending its claim or undercutting its inferences – while $A1$, $A2$ and $A3$ are UND— $A1$ and $A2$ have opposite conclusions and thus attack each other; the conflict is then propagated to the derived argument $A3$.

Let us suppose we require the evaluation of claim b through the `AnswerQuery(Goal)` function in Listing 1. First, the arguments sustaining b are created, in this case only $A1$. Then the evaluation conditions on $A1$ attackers – only $A2$ in this case – are assessed. However, $A2$ admissibility depends, in turn, on $A1$ —as you can see in Figure 1 also $A1$ attacks $A2$. There is a cycle in the graph (Cond2), and no other attackers matching (Cond1). As a consequence, $A2$ is UND and thus $A1$ (Cond3). Accordingly, claim b is labelled UND as expected.

3. Parallelising arguments evaluation

The first issue when facing computational issues of cooperative argumentation is the parallelisation of the argumentation process. Parallelisation needs to be tackled under two distinct perspective: (i) the algorithmic perspective and (ii) the data perspective. Under the algorithmic perspective, we try to divide the argument evaluation (w.r.t. a given semantics) into smaller sub-tasks to be executed in parallel. Under the data perspective, instead, we try to achieve parallelisation by splitting the data used by the algorithm—i.e., the argumentation defeasible theory. Action here is therefore at the data level, looking for possible data partitioning on which the argumentation process can be run in parallel.

Accordingly, in this section we discuss and address both perspectives, respectively in Subsection 3.1 and Subsection 3.2.

3.1. Task parallelisation

Let us consider the algorithm discussed in Subsection 2.2. The purpose of this section is to analyse the requirements and implications of its parallelisation. Note that the part affected to parallelisation is encapsulated in the `Evaluate` function, which is why in the following we take into account that predicate only.

The algorithm structure is simple: the argument evaluation leverages the evaluation obtained from its attackers—i.e., the attackers are recursively evaluated using the same algorithm and

the result is exploited to determine the state of the target argument. Intuitively, a first point of parallelisation can be found in the search and evaluation of the Attackers. Indeed, every condition exploited by the algorithm – (Cond1), (Cond2), and (Cond3) – to evaluate an argument requires one and only one attacker to match the constraint. Those conditions directly suggest an OR parallelisation in the search and evaluation of the attackers. We could evaluate the arguments simultaneously under different branches, and the success in one of the branches would lead to the success of the entire search. Listing 2 shows the modified algorithm.

Listing 2: Evaluate predicate with parallel attackers

```

Evaluate(A, Chain):
  if(PARALLEL {  $\exists B \in \text{Attacker}(A): \text{Evaluate}(B, A \cup \text{Chain}) = \text{IN}$  })
    return OUT
  if(PARALLEL {  $\exists B \in \text{Attacker}(A): B \in \text{Chain}$  })
    return UND
  if(PARALLEL {  $\exists B \in \text{Attacker}(A): \text{Evaluate}(B, A \cup \text{Chain}) = \text{UND}$  })
    return UND
  return IN

```

The algorithm exposes another point of parallelisation. As already suggested, the order in the evaluation of the conditions is essential for the soundness of the algorithm—as illustrated by the following example.

Example 2. *Let us consider argument A and its two attackers B and C . Let it be the case in which we know B and C 's labelling, IN for the former and UND for the latter. If we do not respect the order dictated by the algorithm, A 's labelling is either UND (Cond3) or OUT (Cond1). Of course, the first result would be in contrast with the original grounded semantic requirements for which every argument having an IN attacker should be definitively OUT. Conversely, if we respect the evaluation order, A 's labelling would be OUT in every scenario.*

Although the evaluation order is strict, we can evaluate all the conditions simultaneously and consider the ordering only while providing the labelling for the target argument (mixing AND and OR parallelisation). Listing 3 displays the algorithm modified accordingly. The three conditions are evaluated in parallel, but the result is given accordingly to the defined priorities. If (Cond1) is met, the argument is labelled as OUT. Conversely, even if (Cond2) or (Cond3) are met, one should first verify that (Cond1) does not hold. Only then the argument can be labelled as UND.

Listing 4 contains the final version of the algorithm taking into account both points of parallelisation. The three conditions – (Cond1), (Cond2) and (Cond3) – are evaluated at the same time. Then the results of the three sub-tasks are combined to provide the final solution according to the conditions' priority. Of course, if we consider a scenario where only the first condition (Cond1) is required to determine the status of the argument in input, the parallel evaluation of all the three conditions would lead to a waste of computational resources. However, this problem is easily mitigated by evaluating the sub-task results as soon as they are individually available—i.e. in the case we receive a positive result from a single sub-task, and it is enough to

Listing 3: Evaluate predicate with parallel conditions evaluation

```
Evaluate(A, Chain):
  PARALLEL {
    Cond1 =  $\exists B \in \text{Attacker}(A): \text{Evaluate}(B, A \cup \text{Chain}) = \text{IN}$ 
    Cond2 =  $\exists B \in \text{Attacker}(A): B \in \text{Chain}$ 
    Cond3 =  $\exists B \in \text{Attacker}(A): \text{Evaluate}(B, A \cup \text{Chain}) = \text{UND}$ 
  }
  if(Cond1) return OUT
  if(Cond2 AND NOT Cond1) return UND
  if(Cond3 AND NOT Cond1) return UND
  if(NOT Cond1 AND NOT Cond2 AND NOT Cond3) return IN
```

Listing 4: Evaluate predicate with both parallel conditions evaluation and parallel attackers

```
Evaluate(A, Chain):
  PARALLEL {
    Cond1 = PARALLEL {  $\exists B \in \text{Attacker}(A): \text{Evaluate}(B, A \cup \text{Chain}) = \text{IN}$  }
    Cond2 = PARALLEL {  $\exists B \in \text{Attacker}(A): B \in \text{Chain}$  }
    Cond3 = PARALLEL {  $\exists B \in \text{Attacker}(A): \text{Evaluate}(B, A \cup \text{Chain}) = \text{UND}$  }
  }
  if(Cond1) return OUT
  if(Cond2 AND NOT Cond1) return UND
  if(Cond3 AND NOT Cond1) return UND
  if(NOT Cond1 AND NOT Cond2 AND NOT Cond3) return IN
```

compute the argument status, we can cut the superfluous computational branches and return the final solution.

3.2. Knowledge-base parallelisation

In the first part of our analysis, we focused on the parallelisation problem from a pure computational perspective—i.e., we tried to understand if we can split the evaluation task into a group of sub-task to be executed simultaneously. However, there is another perspective to take into account when parallelising: the one concerning the data.

Example 3. *For example, let us consider a job computing the sum and the product of a set of numbers. Using the sub-task approach, we could have two subroutines running in parallel, one computing the sum and the other computing the product of the numbers. However, leveraging the associativity property of sum and multiplication, we can split the problem into a series of tasks computing both sum and product on a subset of the original data. Then the final result would be the sum and the multiplication of the tasks' results.*

Let us suppose to apply the same principle to the argumentation task. We build arguments from a base theory according to the relations illustrated in Subsection 2.1. The logic theory is, for all intents, the input data of our algorithm (argumentation task). Now, the question is

whether we can effectively split the data into sub-portions to be evaluated in parallel without affecting the global soundness of the original algorithm.

Naïve principle. Let us start with a naïve solution in which we randomly split the input theory between all the available nodes. Of course, this would lead to evident contradictions.

Example 4. For instance, let us consider the following theory (left) and its monolithic evaluation

$$\begin{aligned} r1 & : && \Rightarrow a \\ r2 & : &a & \Rightarrow b \\ r3 & : && \Rightarrow b \\ r4 & : && \Rightarrow \neg a \end{aligned}$$

according to grounded semantic leading to four arguments (right):

$$\begin{aligned} A0 & : && \Rightarrow_{r1} a \\ A1 & : &A0 & \Rightarrow_{r2} b \\ A2 & : && \Rightarrow_{r3} b \\ A3 & : && \Rightarrow_{r4} \neg a \end{aligned}$$

where $A0$, $A1$ and $A3$ are labelled **UND** – since $A0$ and $A3$ attack each other and $A3$ attacks $A1$ – and $A2$ is labelled **IN**. If we leverage a random split, we could have a scenario in which we partition the theory into four parts. Of course, this would lead to a missing argument. Indeed, rules $r1$ and $r2$ are both necessary to conclude $A1$.

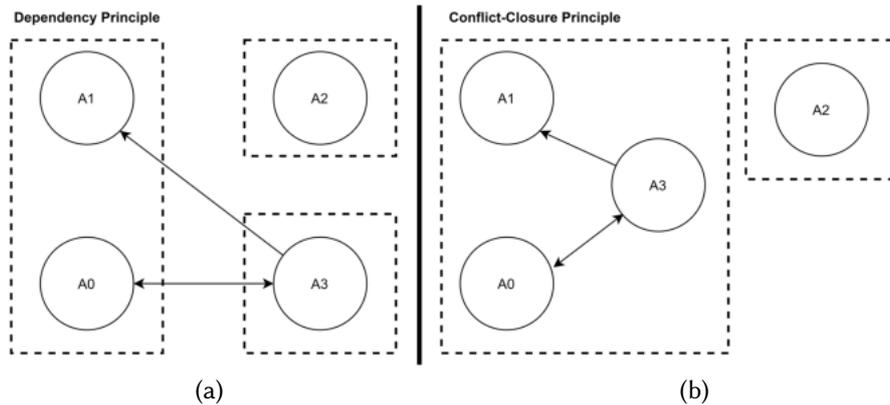


Figure 2: Argumentation graphs and arguments from Example 4 grouped according to dependency (a) and conflict-closure principles (b).

Dependency principle. Now, let us consider a smarter splitting principle based on rules dependency—i.e., if two rules can be chained, they must stay together.

Example 5. Accordingly, if we consider the theory from example 4, we have three subsets of the theory: $r1$ and $r2$, $r3$, $r4$. The evaluation of these three theories would lead to the admissibility of all the four arguments, making the result unsatisfactory w.r.t. the original solution—see Figure 2 (a).

Conflict-closure principle. Observing the abstract argumentation graph it is easy to understand that we cannot split rules claiming conflicting knowledge—see Figure 2 (b). Accordingly, we can observe that a safe split can be guaranteed if the graph-connected sub-portions maintain their integrity—i.e., attacker and attacked arguments belong to the same set.

Example 6. *If we apply this principle to the theory in Example 4, we obtain two sub-portions of the original logic theory allowing for a simultaneous evaluation: r_1 , r_2 and r_4 (set KB_a), and r_3 (set KB_b). The application of the argument evaluation algorithm (in Subsection 2.2) to check the admissibility of b leads to two results: b (A1) is UND (set KB_a), and b (A2) is IN (set KB_b)—coherent with the semantics (Figure 2 (b)).*

Accordingly, in order to guarantee a sound evaluation w.r.t. the original algorithm (Listing 1) the *conflict-closure* principle should be considered while splitting the knowledge base.

4. The complete model

In this section, a complete and sound mechanism for the admissibility task in a fully-concurrent way is provided, exploiting the insights from Section 3 and applying them to an actor-based model [9].

In short, the actor model is based on a set of computational entities – the actors – communicating each other through messages. The interaction between actors is the key to computation. Actors are pure reactive entities that only in response to a message can:

- create new actors;
- send messages to other actors;
- change their internal state through a predefined behaviour.

Actors work in a fully-concurrent way – asynchronous communication and message passing are fundamental to this end – making the actor model suited to concurrent applications and scenarios. We choose this model for its simplicity: it presents very few abstractions making it easy to study both how to model a concurrent system and its properties. The final goal of this research is to provide a sound model for agents’ cooperative argumentation in MAS. Since it is an articulated goal, coping with different dimensions – distribution, sociality, coordination, autonomy – we carry on our investigation in two distinct steps: (1) first, we enable concurrent evaluation of the argumentation algorithms (focusing on distribution), (2) then, we make available the new computational tool in a MAS context (focusing on sociality, coordination, and autonomy). The actor paradigm is a natural choice for the first step of the analysis.

The proposed model embraces both the parallelisation approaches seen in Section 3—i.e., the parallel evaluation of attackers (task parallelisation, Subsection 3.1) and the partitioning of the initial logical theory (data parallelisation, Subsection 3.2).

Listing 5: Master Actor for knowledge base distribution

```
MasterActor:

State:
  responseList
  elem

OnMessage(sender, message):
  if message = AddTheoryMember(term)
    send(ALL, NewTheoryMember(term))
    responseList = []
    elem = term
  if message = Ack(chainingDetails)
    responseList += Ack(chainingDetails)
    evaluateResponses(responseList)
  if message = Nack()
    responseList += Nack()
    evaluateResponses(responseList)

evaluateResponses(responseList):
  if NOT ALL RESPONSE ARE PRESENT:
    return
  if NO ACK IS PRESENT:
    createNewActor(actor)
    send(actor, CreateKnowledgeBase(elem))
  if CONVERGING KB ARE PRESENT:
    selectMergeTarget(target)
    FOR x IN createMergeList(responseList):
      send(x, MergeTheory(target))
```

4.1. Actor-based evaluation: distributing the knowledge base

Let us start with the portion of the model devoted to logic theory distribution. As seen in Subsection 3.2, the adherence to the *Conflict-closure* principle is required to guarantee the soundness of the evaluation of a fragmented theory. However, this observation applies only w.r.t. algorithm in Listing 1 being the arguments' construction and the search for attackers executed on the same task. If we consider instead the concurrent version of the algorithm (Subsection 3.1), the search and evaluation of the attackers are performed in distinct sub-tasks. As a consequence, there is no task required to know how to build an argument and its attacker – the search is delegated to another process –, thus the *Conflict-closure* principle can be ignored. Indeed, a single task in charge of evaluating an argument needs only the portion of the theory required to infer the argument itself—i.e., i.e., only the *Dependency* principle must be respected.

The same idea applies to the actor-based model presented in this section: since the search for attackers is executed concurrently by a set of actors, we require only the *Dependency* principle to be respected.

Since the actor model focuses on actors and their communication the following design will review the structure and behavior of the involved actors. Although a fully distributed version of

Listing 6: Worker Actor for knowledge base distribution

```
WorkerActor:

State:
  kb

OnMessage(sender, message):
  if message = CreateKnowledgeBase(term)
    kb = term
  if message = NewTheoryMember(term)
    if isChainable(term, kb):
      send(sender, Ack(chainingDetails))
      kb += term
    else:
      send(sender, Nack())
  if message = MergeTheory(target)
    send(target, Kb(kb))
    exit
  if message = Kb(newKb)
    kb += newKb
```

the model is possible, we choose to adopt a master-slave approach to simplify the functioning of the system as much as possible.

Two main types of actors are conceived in the system: master (Listing 5) and worker (Listing 6). Master actors coordinate the knowledge base distribution phase while the workers hold a portion of the theory. Accordingly, masters' internal state contains a reference to the term to distribute (`elem`) and a list of the feedbacks from the workers' actors on `elem` distribution (`responseList`), while workers' internal state is simply represented by the portion of the theory they manage, identified by `kb`.

Messages that masters and workers can exchange are represented by the following types:

- `CreateKnowledgeBase`, the first message sent from the master to a new worker containing its initial knowledge base;
- `NewTheoryMember`, sent from the master to all the available workers, through which the master sends the new theory member to be stored in the workers' `kb`;
- `Ack`, sent from a worker to its master in response to a `NewTheoryMember` message, confirms the storing of the new rule in the worker's `kb`;
- `Nack`, sent from a worker to its master in response to a `NewTheoryMember` message, denies the storing of the new rule in the worker's `kb`;
- `MergeTheory`, sent from the master to a set of workers in the case of overlapping theories, orders the workers to conclude their execution after sending their knowledge bases to a targeted worker;

- Kb, sent from a worker A to another worker B, contains the knowledge base that B should add to its own.

If the master receives the order to add a new element to the theory (AddTheoryMember message), three possible scenarios can be configured:

1. none of the workers contains a compatible knowledge base – i.e., it is not possible to chain the new rule to the knowledge base (isChainable returns false) – and consequently, the master creates a new worker containing the portion of the theory (createNewActor);
2. one or more workers have a compatible knowledge base (isChainable returns true), and they add the element to their kb;
3. a set of workers possess overlapping knowledge bases – i.e. the union set of workers' knowledge bases can be used to create a unique inference chain –, and as a consequence, we merge their knowledge bases and destroy the extra workers (MergeTheory message);

Since actors are reactive entities, in order to completely adhere to the actor model the master knowledge base can be changed from outside the actor system—we instruct the master actors to modify the theory through the message AddTheoryMember.

Example 7. *Let us consider again the theory in Example 1. Let us assume a single MasterActor and the following order in the inclusion of the rules in the system: r_1, r_3, r_4, r_2 .² As for the first three rules, the behaviour is the same: the MasterActor issue a NewTheoryMember and receives back only Nack messages—since the rules are not chainable. Accordingly, it creates three distinct workers and sends a single rule to every one of them via the CreateKnowledgeBase message. We now have Worker 1, Worker 2, and Worker 3 with respectively r_1, r_3 and r_4 in their knowledge bases. Then the master issues a NewTheoryMember for r_2 , and both workers 1 and 3 answer with an Ack. Rule r_2 is, in fact, the missing link in the inference chain of r_1 and r_4 . As a consequence, the Master orders a migration to one of them – let us assume Worker 3 – with the MergeTheory message. Worker 3 receives the message, sends its kb to Worker 1 via the Kb message, then stops. At the end of the distribution phase, we have two workers, one containing r_1, r_2, r_4 , the other just r_3 . The dependency principle is thus respected.*

4.2. Actor-based evaluation: evaluating an argument

Let us proceed with the actor-based evaluation of an argument. For this task, we only need one type of actor—WorkerActor in Listing 7. In the final model, we consider workers from Listing 6 and Listing 7 as the same entity. We can evaluate an argument through workers only after they split the logic theory among them according to the mechanism in Subsection 4.1.

Each actor is responsible for evaluating those arguments that can be build using its portion of the theory. When the actor receives an evaluation request, it first checks if attackers exist, w.r.t. its portion of the knowledge base. Then the actor can: (i) register the impossibility to evaluate the argument – only if a cycle through the evaluation chain is detected –, (ii) require the attacker arguments evaluation to all the other actors. In the latter case, the actor shall

²The inclusion order only affects the steps required to converge, not the final state of the system.

Listing 7: Worker Actor for argument evaluation task

```
WorkerActor:

State:
  targets

OnMessage(sender, message):

  if message = Evaluate(claim):
    if buildArgument(claim, arg):
      send(ALL, Attacker(arg, []))

  if message = Attacker(arg, chain):
    if NOT buildAttacker(arg):
      send(sender, In(arg))
    else:
      for attacker IN buildAttacker(arg):
        if attacker IN chain:
          targets += (arg, attacker, sender, [], Und(arg))
        else:
          send(ALL, Attacker(attacker, chain + [arg]))
          targets += (arg, attacker, sender, [], None)
      evaluateResponses()

  if message = Und(arg) OR Out(arg) OR In(arg):
    evaluateResponses(arg)

evaluateResponses(arg):
  for arg, attacker IN targets:
    if ANY OUT:
      targets[attacker] = Out(arg)
    if ANY Und AND NOT ANY OUT:
      targets[attacker] = Und(arg)
    if ALL In:
      targets[attacker] = In(arg)
  if attackersEvaluated(arg):
    sendResponse(sender)
```

answer the original evaluation request only after receiving a response from others actors. The conditions to match while evaluating an argument are the same as the original algorithm in Listing 1:

- if one counterargument is found admissible, we evaluate the argument as OUT;
- if any number of actors decide for the argument undecidability with none advancing its rejection, we mark the argument as UND;
- if all the actors agree that no counterarguments can be provided as acceptable, we evaluate the argument as IN;

Actors provide their suggestions on the state of the requested argument according to all the labels of their counterarguments.

The messages exchanged among worker actors are:

- `Evaluate`, sent to workers (from outside) to require the evaluation of a claim;
- `Attacker`, sent from a worker to all other workers, requires the evaluation of an argument;
- `Und`, `Out`, `In` – sent from a worker to another worker in response to the `Attacker` message – answering the evaluation request.

Note that the `Evaluate` message comes from outside the actor system and starts the evaluation process. In Listing 7, we omit the details on the collection of the `Evaluate` responses and the return of the final result for the sake of conciseness.

Example 8. *Let us continue the example from 1 and 7 and require the evaluation of claim b . From outside the actor system, we send an `Evaluate` message to all the actors. Worker 1 succeeds in building an argument ($A1$) and sends to all the other workers – also Worker 1 is included in the list – an `Attacker` message requiring attackers evaluation. Worker 1 answers with an `In` message – there are no attacking arguments according to its knowledge –, while Worker 2 sends back an `Und` response. Indeed, worker 2 is able to create a valid counterargument ($A2$), but a cycle is detected in the inference chain. According to the evaluation algorithm, receiving an `Und` and an `In` as a response, Worker 1 can finally label $A1$ as `UND`.*

5. Related & Conclusions

The work presents a first approach to the problem of cooperative argumentation in the context of a MAS. Starting from the single query evaluation mode of `Arg2P` – aimed at evaluating the admissibility of a single statement without the need to build the entire argumentation graph – we introduce the corresponding distributed computational model. We first discuss how the argument evaluation algorithm of `Arg2P` can be parallelised, then we deliver a complete model for decentralised reasoning based on the actor model.

Our work follows the insights from the ones in [10] and [11, 12]. The former has been the first proposal of a tool – also based on the `tuProlog` system – exploiting a dialogical argumentation mechanism—i.e., argumentation is performed across multiple processes proposing arguments and counterarguments. However, the argumentation algorithm distribution has not been addressed. Conversely, in [11, 12] the authors directly address the problem of enabling argumentation techniques in MAS. Nonetheless, their technique exploits a centralised evaluation of all the knowledge spread across MAS agents, thus exposing serious problems to the scalability of their approach.

Our work can be extended in various directions. First, we shall provide an implementation of the distributed model in the `Arg2P` framework. In fact, in this work, we discuss the parallelisation problem from a theoretical perspective. Only once implemented the approach, it will be possible

to compare the performances of the monolithic and distributed versions of the algorithm properly addressing a discussion on efficiency and scalability issues.

Then, a well-founded analysis of the model is still missing, discussing formal properties such as soundness and completeness. Moreover, experiments need to be run in a MAS environment. There, the open issues are many, e.g., how could agents benefit from this mechanism? How does coordination media impact the model?

Finally, it is worth highlighting that in this work we distribute the knowledge base across actors in order to maximise the scalability of the system. The consequences of using the model in a context where the nodes possess an arbitrary knowledge – as agents in MAS – are still to be inspected.

Acknowledgments

G. Pisano and R. Calegari have been supported by the H2020 ERC Project “CompuLaw” (G.A. 833647). A. Omicini has been supported by CHIST-ERA project “Expectation” (G.A. CHIST-ERA-19-XAI-005).

References

- [1] H. Jung, M. Tambe, S. Kulkarni, Argumentation as distributed constraint satisfaction: Applications and results, in: 5th International Conference on Autonomous Agents (Agents '01), 2001, pp. 324–331. doi:10.1145/375735.376322.
- [2] Á. Carrera, C. A. Iglesias, A systematic review of argumentation techniques for multi-agent systems research, *Artificial Intelligence Review* 44 (2015) 509–535.
- [3] G. Pisano, R. Calegari, A. Omicini, G. Sartor, Arg-tuProlog: A tuProlog-based argumentation framework, in: F. Calimeri, S. Perri, E. Zumpano (Eds.), CILC 2020 – Italian Conference on Computational Logic. Proceedings of the 35th Italian Conference on Computational Logic, volume 2710 of *CEUR Workshop Proceedings*, Sun SITE Central Europe, RWTH Aachen University, CEUR-WS, Aachen, Germany, 2020, pp. 51–66. URL: <http://ceur-ws.org/Vol-2710/paper4.pdf>.
- [4] R. Calegari, G. Contissa, G. Pisano, G. Sartor, G. Sartor, Arg-tuProlog: a modular logic argumentation tool for PIL, in: S. Villata, J. Harašta, P. Křemen (Eds.), Legal Knowledge and Information Systems. JURIX 2020: The Thirty-third Annual Conference, volume 334 of *Frontiers in Artificial Intelligence and Applications*, 2020, pp. 265–268. doi:10.3233/FAIA200880.
- [5] S. Modgil, H. Prakken, The ASPIC⁺ framework for structured argumentation: a tutorial, *Argument Computation* 5 (2014) 31–62. doi:10.1080/19462166.2013.869766.
- [6] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence* 77 (1995) 321–358. doi:10.1016/0004-3702(94)00041-X.
- [7] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, *The Knowledge Engineering Review* 26 (2011) 365–410. doi:10.1017/S0269888911000166.

- [8] A. J. García, G. R. Simari, Defeasible logic programming: An argumentative approach, *Theory and Practice of Logic Programming* 4 (2004) 95–138. doi:10.1017/S1471068403001674.
- [9] C. Hewitt, P. B. Bishop, R. Steiger, A universal modular ACTOR formalism for artificial intelligence, in: *3rd International Joint Conference on Artificial Intelligence*, William Kaufmann, 1973, pp. 235–245. URL: <http://ijcai.org/Proceedings/73/Papers/027B.pdf>.
- [10] D. Bryant, P. J. Krause, G. Vreeswijk, Argue tuProlog: A lightweight argumentation engine for agent applications, in: *Computational Models of Argument*, volume 144 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2006, pp. 27–32. URL: <http://www.booksonline.iospress.nl/Content/View.aspx?piid=1922>.
- [11] E. Oliva, P. McBurney, A. Omicini, Co-argumentation artifact for agent societies, in: S. Parsons, I. Rahwan, C. Reed (Eds.), *Argumentation in Multi-Agent Systems*, volume 4946 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 31–46. doi:10.1007/978-3-540-78915-4_3, 4th International Workshop (ArgMAS 2007), Honolulu, HI, USA, 15 May 2007. Revised Selected and Invited Papers.
- [12] E. Oliva, M. Viroli, A. Omicini, P. McBurney, Argumentation and artifact for dialogue support, in: I. Rahwan, P. Moraitis (Eds.), 5th International Workshop “Argumentation in Multi-Agent Systems” (ArgMAS 2008), AAMAS 2008, Estoril, Portugal, 2008, pp. 24–39. doi:10.1007/978-3-642-00207-6_7.

SW-CASPAR: Reactive-Cognitive Architecture based on Natural Language Processing for the task of Decision-Making in the Open-World Assumption

Carmelo Fabio Longo¹, Corrado Santoro¹, Domenico Cantone¹,
Marianna Nicolosi-Asmundo¹ and Daniele Francesco Santamaria¹

¹Department of Mathematics and Computer Science, University of Catania, Viale Andrea Doria, 6, 95125 Catania, Italy

Abstract

This paper addresses the issue of nowadays vocal assistants cognitive lacks, which are able to execute from vocal commands only simple plans without higher capabilities of decision-making. In this work we propose an *open-world assumption* transposition of the cognitive architecture CASPAR, whose heuristic takes into account of meta-reasoning in the *closed-world assumption*, namely SW-CASPAR. Such a cognitive architecture is also provided with a module for semi-automatic *ontology learning* from sentences in natural language, reflecting the domain with an instance of a novel foundational ontology called Linguistic Oriented Davidsonian Ontology (LODO), with the aim of increasing the deepness of reasoning without compromising linguistic-related features. LODO is inspired by the *First-Order Logic Davidsonian* notation and serialized in OWL 2. A case-study applied to automation on health scenarios is also provided.

Keywords

Cognitive Architecture, Natural Language Processing, Artificial Intelligence, Semantic Web, Internet of Things, Ontology Learning, Computational Linguistic

1. Introduction

In the last years, the market of the Internet of Things (IoT) has become quite disruptive by changing the lifestyle of thousands of people. Thanks to a considerable number of sensors and actuators interconnected with each other, a great number of environments, both home and industrial, have been enriched with every kind of automation, whose diversity is bounded only by the imagination of the designer. In most of such environments, especially homes, vocal assistants, together with the rest of the compatible devices, assume a more and more important role. The two major enterprises in the market of vocal assistants, namely *Amazon* and *Google*, by leveraging their massive dataset for training models in the task of vocal commands recognition, aim more at increasing their product's pervasiveness than respect to improving native reasoning capabilities of the latters. In both cases, each vocal command can be related either to a single pair or to a group of request/operation triggered by specific words. Additional features can be

WOA 2021: Workshop "From Objects to Agents", September 1–3, 2021, Bologna, Italy

✉ fabio.longo@unict.it (C. F. Longo); santoro@dmi.unict.it (C. Santoro); domenico.cantone@unict.it (D. Cantone); nicolosi@unict.it (M. Nicolosi-Asmundo); santamaria@dmi.unict.it (D. F. Santamaria)

🆔 0000-0002-2536-8659 (C. F. Longo); 0000-0003-1780-5406 (C. Santoro); 0000-0002-1306-1166 (D. Cantone); 0000-0003-4456-5110 (M. Nicolosi-Asmundo); 0000-0002-4273-6521 (D. F. Santamaria)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

included by developers with the so-called “Skills” or by exploiting external cloud services like IFTTT [16], but with no enhanced reasoning capabilities.

With “enhanced reasoning capabilities” we intend not only the ability to infer the proper association *command*→*plan* from utterances, but also to combine facts with rules in order to infer new knowledge and help the user in decision-making tasks. In order to help users in their cognitive processes by using a form of logic reasoning, we must provide the assistant with a combination of facts and rules or, more simply, we must give assistants the capabilities of freely and implicitly extracting what they require from texts in natural language.

In a prior work [1, 2], some of the authors of this paper designed a cognitive architecture called *CASPAR* that instantiates cognitive agents provided with both reactive and cognitive reasoning. Such agents are able to reason in a process subordinated by a further level of reasoning, namely *meta-reasoning*, in a conceptual space, whose content is made of facts and axioms in first-order logic with the closed-world assumption. An important variation of such architecture, which would make it suitable for different scenarios and represents the main motivation of this paper, consists in reasoning over shared *ontologies* in the open-world assumption, in other words by leveraging the *Semantic Web*.

Ontologies are formal, explicit specification of a shared conceptualization [3]. The Semantic Web, with all its layers and frequent updates, can be considered the very backbone of nowadays ontologies.

The closed-world assumption applies when a system has complete information, like many database applications. On the contrary, the open-world assumption applies when a system has incomplete information. For example, consider a patient’s clinical history system. If the patient’s clinical history does not include a particular allergy, it would be incorrect to state that the patient does not suffer from that allergy; it is unknown if the patient suffers from that allergy, unless more information is given to disprove the assumption.

On the basis of the above, we designed a variation of *CASPAR*, namely *SW-CASPAR*¹, capable of parsing IoT commands from natural language, subordinating them by a meta-reasoning on the Semantic Web. An important effort in the development of such architecture has been given by the *Owlready* [4] library, which gives also the chance of reasoning in the *local closed world*², as reported in [5].

SW-CASPAR relies on the support of a module called *Ontology Builder*, which dynamically creates domain-legacy ontologies serialized in OWL 2, in order to allow for meta-reasoning. Nonetheless, it can be used also as an alternative stand-alone tool for creating ontologies, in substitution of other state-of-the-art tools, for several reasons that will be clarified in later on. A Python prototype implementation of *SW-CASPAR* is also available as a Github repository.³

This paper is structured as follows: Section 2 illustrates the issues of ontology learning from natural language; Section 3 describes the state of the art of related literature; Section 4 shows in detail all the architecture’s components and underlying modules, whereas Section 5 illustrates the strategy applied for the task of the Ontology Learning; Section 6 depicts a case-study of an agent working on a health scenario, making usage of both reasoning and meta-reasoning in the

¹Which stand for: *Semantic Web-CASPAR*.

²*Owlready* reasoning capability in the *local closed world* is limited to a set of individuals and classes.

³<http://www.github.com/fabiuslongo/sw-caspar>

Semantic Web; finally, in Section 7 we draw our conclusions and mention some future work perspectives.

2. The issue of Natural Language Ontology

The task of learning an ontology reflecting a domain, by means of a description in natural language of the domain itself, implies several bias that must be considered. In general, ontology learning can be accomplished in three ways: manual learning, by leveraging the effort of experts of the domain; cooperative learning, where most or all the task are supervised by experts; semi-automatic learning, where the ontology construction process is performed automatically with limited intervention by users or experts. It is worth mentioning that full automatic learning by a system is still a significant challenge and it is not likely to be feasible [6]. When sentences are not specifically well-formed, the task of ontology learning from natural language can be quite hard, because of all possible semantic ambiguities of idioms and the arbitrary descriptive nature of the world, which can induce morphologically distinct sequences of words to express the same concept.

In [7], it is reported that “Natural Language Ontology is a branch of both metaphysics and linguistic semantic. It aims to uncover the ontological categories, notions and structures which are implicit in the use of natural language, that is, the ontology that a speaker accepts when using a language”. But such an acceptance implies several issues to be addressed by an ontology learning system designer. For instance, for a speaker it would be quite natural and simple to express that a certain object does not exist. In a closed-world assumption we can limit to not assert such a concept, or at least to retract the representation of it from a knowledge base. But how to operate in an open-world assumption, in order to let such information participate in a reasoning process in a human-like fashion? How to keep consistency when an information and its complement are possibly both present in an ontology, as it could happen in texts given by a speaker? In linguistic science, intentional objects as *nonexistent* are considered particularly problematic [8]; for instance, having an ontology A representing a domain of existing entities, and another ontology B representing a description in natural language of the same domain, we cannot definitively say that A and B are equivalent, due to a possible introduction in B of entities which not exist but are functional to the arbitrarily descriptive use of words present in the source text of B.

Let us consider the following two sentences: “*Robert walked down the street*” and “*Robert had a walk down the street*”. How can one infer that they express the same concept in a decision process? In the second sentence we are in presence of the so-called *deverbal nominalization* of the verb *walk* versus a noun expressing the action of walking. So we are forced to create somehow a bridge between the two sentences, in order to achieve the same result when both are participating in a reasoning process. Similarly, let us consider also the simple verbal phrase “*the friends are happy*” and the snippet “*happy friends*”. In this case, we are in presence of a *deadjectival nominalization*, because the adjective *happy* becomes the object of a copular⁴ verb

⁴A copular verb is a special kind of verb used to join an adjective or noun complement to a subject. Common examples are: be (is, am, are, was, were), appear, seem, look, sound, smell, taste, feel, become, and get. A copular verb expresses either that the subject and its complement denote the same thing or that the subject has the property

(be), thus the subject (friends) has the property denoted by the adjective; hence, how to express, by means of an ontology, that the two pieces of information yield the same concept?

The above questions and many other issues related to the natural language ontology are described in [7], which can be a good starting point for filling the gap between a domain and an arbitrary description of it given by a speaker.

In [1, 2], the authors gave an overview of CASPAR, from which SW-CASPAR inherits most of its features, so in the next section we limit ourselves to highlight the state-of-the-art concerning ontology learning, whereas motivations and results concerning the road we chose to follow for the design of this architecture's ontology learning system are postponed until Section 5.

3. Related works

The architecture explained in this paper inherits all the *cognitive* IoT features from its predecessor, so the reader is referred to [1, 2] for more details. Hence, this section is focused more on the scope of ontology learning from natural language, which is the main additional contribution of this work.

The disruptive growing of textual data on the web, coupled with an increasing trend to promote the semantic web, has made the automatic ontology construction from texts a very promising research area. However, manual construction of ontologies is time consuming as well as an extremely laborious and costly process. For this reason, several approaches have been designed to automatize the ontology learning from text, each with different levels of human interaction. Such approaches can be divided into two categories: linguistic-based and machine learning approaches. Among the linguistic-based approaches, the authors of [9] use semantic templates and lexico-syntactic patterns such as "NP is type NP" to extract hypernym and meronym relations. But it is well known that these approaches have reasonable precision, though they have a very low recall [6]. In order to achieve terms extraction, [10] leverages POS tagging to assign parts-of-speech to each word and a rule-based sentence parser. However, many words are ambiguous and so this approach will lead to a low accuracy, without a valid disambiguation strategy. Our approach, although similar, makes usage also of a performative disambiguation module, described in details in [2], and extracts also conditional-word based axioms. The authors of [11] make use of a dependency parser to map syntactic dependencies into semantic relations. Such approaches are useful for terms and concepts extraction and also for relations discovery, even though they need to cooperate with other algorithms and/or rules for better performance.

As for machine learning approaches, the system ASIUM [12] adopts agglomerative clustering for taxonomy relations discovery. The axioms only express subsumption relationship (IS-A) between unary predicates and concepts. The system OntoLearn [13] extracts only taxonomic relations, taking into account hypernyms from WordNet. The system HASTI [14] builds automatically ontologies from scratch, using logic-based, linguistic-based and statistical-based approaches. It is one of the few systems that try to learn axioms using inductive logic programming, even though they are very general. Furthermore, such a system has the limitation that

denoted by its complement.

each intermediate node, in the conceptual hierarchy, has at most two children. Worth of mentioning, there is also Text-to-Onto [15], which builds taxonomic and non-taxonomic relations that make use of data mining and natural language processing. Other approaches [16, 17, 18] are also interesting, although either they have limitations on the composition of learned concepts or they generate too many hypotheses, making the involved calculation unmanageable.

Besides a large analysis of state-of-the-art, the authors of [19] discuss on reasons and techniques about the usage of deep neural networks in the Ontology Learning. In these cases, neural networks are often hard to train, although in many cases they give better results by using large, domain-related datasets.

4. The Architecture

As pointed out in the introduction, the architecture's name derives directly from its predecessor CASPAR, namely SW-CASPAR, which stands for **Semantic Web-Cognitive Architecture System Planned and Reactive**, a name that summarizes its inherited features plus the transposition on the Semantic Web. All interacting components are depicted in Fig. 1, highlighted with distinct colours.

The main component of this architecture, namely the *Reactive Reasoner*, acts as “core router” by delegating operations to other components, and providing all needed functions to make the whole system fully operative.

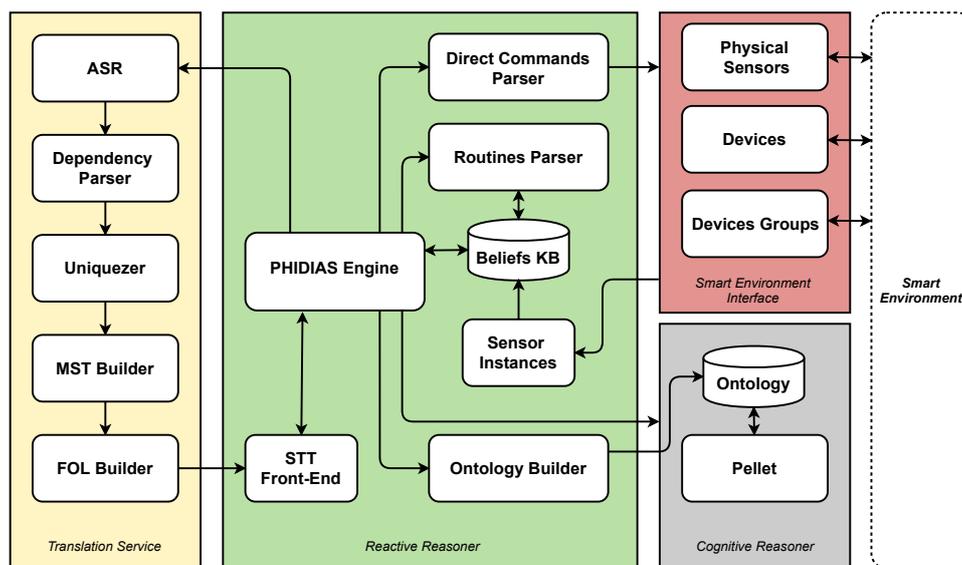


Figure 1: The Software Architecture of SW-CASPAR.

This architecture's knowledge base (KB) is divided into two distinct parts operating separately, which we will distinguish as *Beliefs KB* and *Ontology*: the former contains information about physical entities which affect and are affected on the agent, whereas the latter contains conceptual information not directly perceived by agent's sensors, but on which the agent would

make inferences.

The Beliefs KB provides exhaustive cognition about what the agent could expect as input data coming from the outside world; as the name suggests, such a KB is fed by specific beliefs that can - in turn - activate related plans in the agent's behaviour.

The Ontology is defined by triples in OWL 2, and is fed by the Ontology Builder within the Reactive Reasoner.

The two KBs represent, somehow, two different kinds of human being memory: the so-called *procedural memory* or *implicit memory*[20], made of thoughts directly linked to concrete and physical entities, and the *conceptual memory*, based on cognitive processes of comparative evaluation.

As well as in human beings, in this architecture the two KBs can interact with each other in a very reactive decision-making process (meta-reasoning).

4.1. The Translation Service

The *Translation Service* component (left box in Figure 1) is a pipeline of five modules with the task of taking a sound stream in natural language and translating it in a *neo-davidsonian* FOL expression inheriting the shape from the event-based formal representation of Davidson [21]. The reader is referred to [1, 2] for a detailed description of such a component.

4.2. The Reactive Reasoner

This component (central box in Fig. 1) has the task of letting other modules communicate with each other; it also includes additional modules such as the Speech-To-Text (STT) Front-End, which transforms information coming from other modules in beliefs, IoT Parsers (Direct Command Parser and Routine Parser), Sensor Instances, and Ontology Builder. The Reactive Reasoner contains also the Beliefs KB, which supports both Reactive and Cognitive reasoning.

The core of this component processing is managed by the Belief-Desire-Intention Framework Phidias [22], which gives Python programs the ability to perform logic-based reasoning (in Prolog style) and lets developers write reactive procedures, i.e., pieces of programs that can promptly respond to environment events. For further details about such a component, readers are referred again to [1, 2].

4.3. The Smart Environment Interface

This component (upper right box in Fig. 1) provides a bidirectional interaction between the architecture and the outer world. In [23], we have shown the effectiveness of this approach by leveraging the Phidias predecessor Profeta [24], even with a shallower analysis of the semantic dependencies, as well as an operation encoding via WordNet [25] in order to make the operating agent multi-language and multi-synonymous. For more details, the reader is still referred to [1, 2].

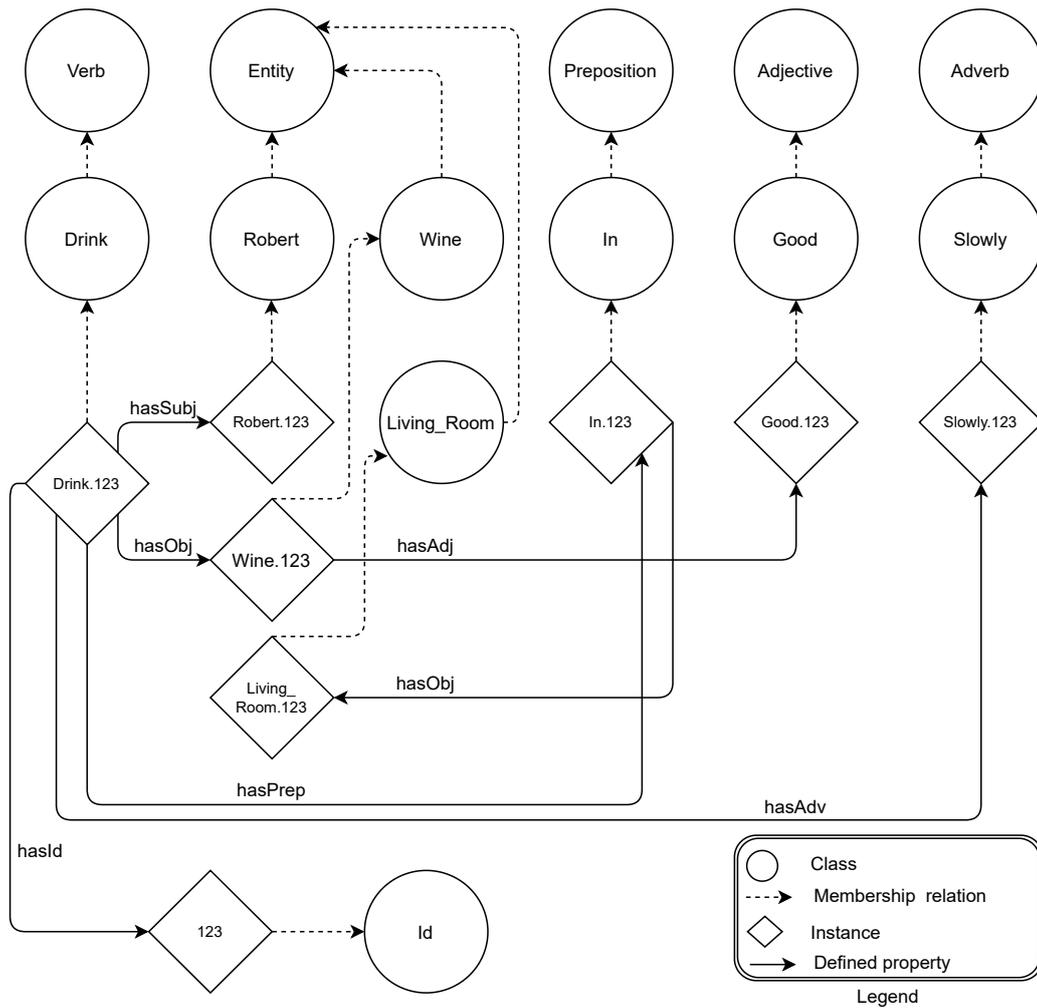


Figure 2: A simple instance of LODO ontology related to the sentence (2).

4.4. The Cognitive Reasoner

This component (bottom right box in Figure 1) allows an agent to implicitly invoke the Pellet reasoner at runtime, in order to achieve a meta-reasoning subordinating agent's IoT commands. This component comprises the ontology as well, which is fed by the Ontology Builder within the Reactive Reasoner.

5. The Ontology Learning

Differently from most approaches to ontology learning, in this paper we give up the idea that whatever such approach will be, it will suffer from the biases related to the ontology of natural language. In light of that, firstly we create a First-Order Logic representation directly linked to

the linguistic features of the sentences in exam, which is provided by the Translation Service component; secondly, whether required, we provide the ontology-specific SWRL [26] rules, whom extend the OWL 2 expressiveness by adding Horn-like axioms. Such rules, as we can see next in the paper, will contribute to fill the gap between the ontology itself and the expected reasoning in human-like fashion. The core module of the Translation Service, as depicted in Fig. 1, is the *Macro Semantic Table* (MST) Builder. Since the MST Builder is made of production rules taking into account relations (dependencies) between words, as long as such relations are treated properly by some rules, the accuracy of the conversion from natural language to logical form can be clearly considered equal to the accuracy of the dependency parser. As for the latter, which in this work's case-study is spaCy [27], in the author's website⁵ it is reported to have a state-of-the-art accuracy of 90% for all trained models available for the english idiom.

As reported in detail in [1, 2], the *Translation Service* translates a text in natural language into a First-Order Logic (FOL) expression, inheriting the shape from the event-based formal representation of Davidson [21]. Thus, having such a FOL representation, the module STT Front-End, taking into account the Part-of-Speech which are parts of each label's predicates, will assert specific beliefs triggering the production rules system of the Ontology Builder. The latter has the task of physically creating the OWL 2 domain-legacy ontology containing the triples representing all verbal phrases and their satellite semantic parts (nouns, adjectives, prepositions, and adverbs). For its direct derivation from Davidson notation, in this work we define such a family of ontologies as **LODO** (**Linguistic Oriented Davidsonian Ontology**). The latter can be considered a foundational ontology, i.e., a specific type of ontology designed to model high-level and domain independent categories about the real world.

The general schema of LODO is quite straightforward. We define *regular verbal phrase* a set of triples in OWL 2 made by the following classes and their instances:

- *Verb*. Each instance of this class represents what comes as verbal phrase in the Davidsonian notation, from the Translation Service. Each individual has the following object properties: *hasId*, having the values of a unique timestamp; *hasSubject*, representing the verb subject in the domain of *Entity*; *hasObject*, representing the verb object in the domain of either *Entity* or *Verb* (in the case of embedded verbal actions). Another property, namely *isPassive*, possibly indicates whether a verbal action is passive or not.
- *Id*. Each instance of this class represents a unique timestamp related to a verbal actions. It takes the value of the object property *hasId* from some instance of *Verb*. As in temporal logic, such value can be useful to deal with inconsistency cases: the higher is the Id, the more valid is the related instance of *Verb*, even when such an instance has the property *hasAdverb* equal to the value of *Not*⁶; notice that a proper SWRL axiom could be also used to invalidate such obsolete individuals, in order to let them not participate in a reasoning process. Furthermore, by taking into account the Part-of-Speech, it can also introduced an object property *hasTime* of such instance, to express the tenses of the verbal actions (Present, Past Tense, Past Participle, Gerund) that respect the timestamp.
- *Entity*. Each instance of this class represents an entity referenced by the object property either *hasSubject* or *hasObject*. Compound nouns are concatenated in order to form a

⁵https://spacy.io/models/en#en_core_web_lg

⁶Negations are treated as whatever adverb.

single individual.

- *Adjective*. Each instance of this class takes the values of the object property *hasAdj* of some instance of *Entity*.
- *Preposition*. Each instance of this class represents a preposition and it is referenced by the object property *hasPrep* of some instance of either *Verb* or *Entity*. Moreover, each of such instance has the object property *hasObject* referencing some instance of *Entity*.
- *Adverb*. Each instance of this class represents an adverb and has the values of the object property *hasAdv* of some instance of *Verb*.

Together with such taxonomic and non-taxonomic relations, LODO comprises also a group of axioms (or part of them) implicitly created by SW-CASPAR, with the aim of increasing the chances of reasoning/graph matching. Such axioms are summarized as follows:

- *Assignment Rules*. Such rules are implicitly asserted in the presence of a FOL expression representing a copular verb (possibly identified also by its synset) and its satellite predicates. Formally, in the presence of the following Davidsonin FOL expression coming from the Translation Service in Fig. 1:

$$\text{Subject:POS}(x_1) \wedge \text{Cop:POS}(e_1, x_1, x_2) \wedge \text{Object:POS}(x_2)$$

where each predicate has its own Part-of-Speech (POS) tag. Such a expression will trigger the assertion of the following SWRL rule⁷:

$$\text{Subject}(?x) \rightarrow \text{Object}(?x) \quad (1)$$

That's because a copular verb (such as *Be*, for instance) is an intransitive verb but identifying its subject with its object; hence, in this case, the class membership of the verb's object will be inherited by the subject.

- *Legacy Rules*. Such rules are implicitly asserted together with the *Assignment Rules*, to let a copular verb's subject inherit both adjectives and prepositions properties of the verb's object. Formally, considering (1), the corresponding legacy rule will be the following:⁸

$$\text{Subject}(?x_2), \text{Object}(?x_1), \text{hasAdj}(?x_1, ?x_3), \text{Adjective}(?x_3) \rightarrow \text{hasAdj}(?x_2, ?x_3)$$

- *Deadjectival Rules* (optional). In the presence of an instance of *Adjective*, such rule asserts a new deadjectivated instance of the latter as new membership of the adjective related noun. Formally:

$$\text{Entity}(?x_1), \text{hasAdj}(?x_1, ?x_2), \text{Adjective}(?x_2) \rightarrow \text{Entity}(?x_2)$$

- *Deverbal Rules* (in progress of development). In the presence of an instance of *Verb*, such rules assert a new deverbalized instance of the latter having the same entities as the former.

⁷By omitting the POS, which can be also included in classes labelling.

⁸Similarly for preposition, by changing *hasAdj* with *hasPrep*.

- *Implicative Copular Rules*. Such rules take into account implicative axioms, possibly coming from the Translation Service in FOL Davidsonian notation and containing a single copular verb in the implication's head. They are useful to infer new memberships of the initial sentence subject, which must be present also in the body. The production rule of the Ontology Builder for such rules assertion takes into account the following pattern:

$$\text{Subject}(x_{body}) \wedge \dots \implies \text{Subject}(x_{subj}) \wedge \text{Object}(x_{obj}) \wedge \text{Cop}(e_{cop}, x_{subj}, x_{obj})$$

As shown above, the label Subject must be in both left- and right-hand side of the FOL expression; otherwise, in order to replace possible pronouns that invalidate the pattern, an anaphora resolution pre-processing could be required before the Translation Service pipeline in Fig.1. Cop is the label of a copular verb which will be absorbed, permitting the formal assertion of the following pattern:

$$\text{Subject}(?x_{obj}), \dots \rightarrow \text{Object}(?x_{obj})$$

Any other implicative FOL expression with non-copular verb in the head will be discarded, due to the non-monotonic features of SWRL.

- *Value Giver Statements* (optional). Such a statement contributes to give a value to a data property *hasValue* related to a specified individual, which is parsed by the Ontology Builder by matching the following pattern of beliefs:

$$\text{GND}(\text{FLAT}, X, Y), \text{ADJ}(\text{FLAT}, X, \text{"Equal"}), \text{PREP}(\text{FLAT}, X, \text{"To"}, S), \text{VALUE}(\text{FLAT}, S, V)$$

The first argument (FLAT) of each belief is for distinguishing non-implicative expressions from implicative ones, and even either right- or left-hand side for the latter. The belief GND is related to a ground term with label Y coming from a FOL expression, which corresponds to a couple of class-individual in the ontology. The beliefs ADJ and PREP specify a lexical content among their arguments, while VALUE specifies the value that must be given to the individual corresponding with label Y. The property *hasValue* might be involved in comparison operations in the composition of a SWRL axiom.

- *Values Comparison Conditional* (optional). Such conditionals are parsed from sentences similarly to the *Value Giver Statement*, but they will take place within the body of Implicative Copular Rules.

For instance, let us consider the following sentence:

$$\textit{Robert slowly drinks good wine in the living room.} \quad (2)$$

The Translation Service will give the following FOL expression as result:

$$\text{Robert:NNP}(x_1) \wedge \text{wine:NN}(x_2) \wedge \text{drink:VBZ}(e_1, x_1, x_2) \wedge \text{slowly:RB}(e_1) \wedge \text{good:JJ}(x_2) \wedge \\ \text{in:IN}(e_1, x_3) \wedge \text{living:NN}(x_3) \wedge \text{room:NN}(x_3)$$

Then, on the basis of POS and arguments cardinality, the STT Interface will produce the following set of beliefs:

PREP(FLAT, e1, In, x3), ACTION(ROOT, FLAT, Drink, e1, x1, x2), GND(FLAT, x3, Living),
 GND(FLAT, x1, Robert), GND(FLAT, x2, Wine), GND(FLAT, x3, Room), ADV(FLAT, e1, Slowly),
 ADJ(FLAT, x2, Good)

Since implicative copular rules can be applied only one time for sentence, the label ROOT as first argument of the belief ACTION has the aim of distinguishing the main verbal action from possible others in the same sentence. The final step is done by the Ontology Builder (within the Reactive Reasoner in Fig. 1), whose production rules will match in a specified order such beliefs, in order to interface with the Owlready libraries and create the OWL 2 ontology.

Fig. 2 depicts such an ontology: the classes in the upper level (*Verb*, *Entity*, *Preposition*, and *Adverb*) are meant to be subclasses of *Things*; the remaining ones in the circles are subclasses of the former; the diamond shaped boxes are individuals whose label contains also a reference of verbal action's Id (the value 123 is not indicative); the latter is also an individual itself, being instance of the class *Id*.

Due to the presence of the adjective *good* related to the individual *wine*, optionally one might activate the SW-CASPAR deadjectival generation rule; then, invoking an OWL 2 reasoner such as Hermit [28] or Pellet [29], the individual *wine* will achieve the new deadjectivated membership *good*.

6. Case-Study

In this section we present a simple case of ontology building and reasoning/meta-reasoning, showing how IoT agents based on SW-CASPAR are able to parse natural language commands and reason about their execution in the open-world assumption. Inspired by [5], where the author shows a use-case consisting in reasoning on drugs contraindications in presence of food intolerances, in this case-study we mainly focus on clinical information about a patient's health disorders and known issues of drugs.

We suppose to provide a hospital with a (semi-)automatized drug distribution system based on natural language recognition, or even to build a robot performing such a task. In this scenario, we suppose to define one or more agents assisting physicians in the decision-making task related with the administration of drugs, on the basis of known drug's issues and clinical picture of patients. In order to address such a task, we extended the Smart Environment Interface of SW-CASPAR with the following set of two production rules given by Phidias,⁹ which considers a known issue of the drug *Rinazina* concerning a contraindication for hypertensive patients:

+INTENT("Rinazina", T) / eval_sem(T, "Hypertensive") » [say("Nope. Patient is hypertensive")] (3)

+INTENT("Rinazina", T) » [exec_cmd("Rinazina", T), say("execution successful")] (4)

Each production rule which begins with +¹⁰ is expressed in a Prolog-like syntax, where the left hand-side comprises a belief we want the rule to match with (INTENT), conditioned by

⁹For the sake of the case-study we consider a simplified form of rules. The reader is referred to the Github repository for more details.

¹⁰Thanks to an *ad-hoc* operators override in Python.

other beliefs (in this case the Active Belief *eval_sem*); the right-hand side comprises, in square brackets, the plan to execute when the rule is triggered.

We now generate the domain-legacy ontology exploiting LODO with all the required information about the patient *Robinson Crusoe* and his health disorders starting from the following sentences:

Robinson Crusoe is a patient

Robinson Crusoe has diastolic blood pressure equal to 150

When a patient has diastolic blood pressure greater than 140, the patient is hypertensive

The first sentence is parsed as both regular verbal phrase and assignment rule, whose related classes and individuals are shown in Fig. 3. Classes and instances have similar names (except for the timestamp) by the virtue of *punning patterns*, which increase the chances of reasoning, whereas a unique timestamp is adopted for all the elements of the same verbal phrase.

Together with classes and individuals, the corresponding legacy rules are also asserted (first and third rule of Fig. 4) allowing *Robinson Crusoe* to inherit all the features of the individual *patient* in an analogous way as the speaker's knowledge flow in the scope of the same discourse. The developer might also provide customized IRI,¹¹ either manually or automatically, by means of a pre-compiled association table.

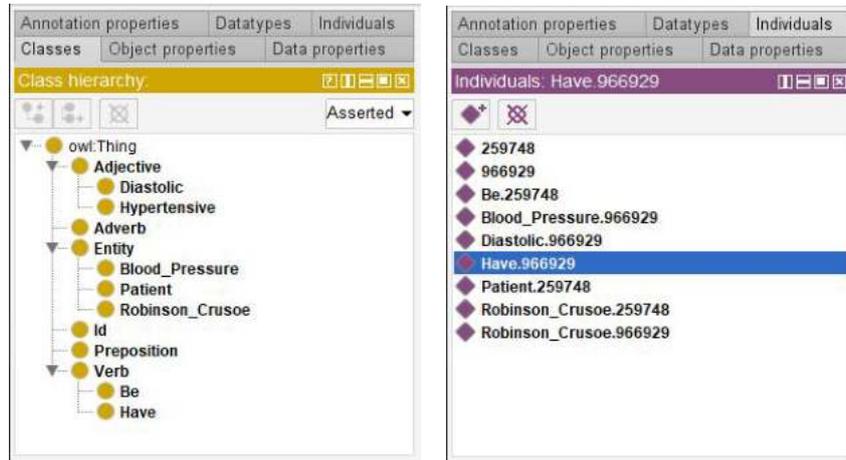


Figure 3: The LODO taxonomic relations and instances of the case-study

As shown in Fig. 4, the second sentence is parsed as regular verbal phrase containing also a value giver statement (depicted in Fig. 5).

The third sentence is parsed by the Translation Service as FOL expression containing an implication such as:

$$\text{Have:VBZ}(e_1, x_1, x_2) \wedge \text{Blood:NN}(x_2) \wedge \text{Patient:NN}(x_1) \wedge \text{Pressure:NN}(x_2) \wedge \text{Than:IN}(x_2, x_5) \\ \wedge \text{Diastolic:JJ}(x_2) \wedge \text{Great:JJR}(x_2) \wedge 140:\text{CD}(x_5) \implies \text{Patient:NN}(x_3) \wedge \text{Hypertensive:JJ}(x_4) \wedge \\ \text{Be:VBZ}(e_2, x_3, x_4)$$

¹¹Internationalized Resource Identifier



Figure 4: The LODO rules of the case-study

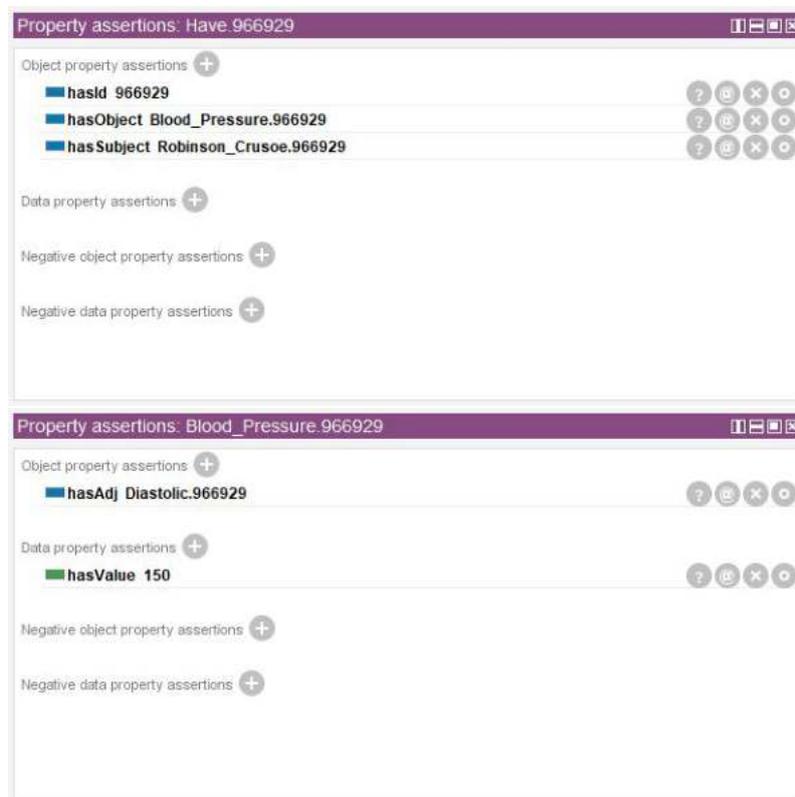


Figure 5: The LODO non-taxonomic relations of the case-study

Then the Ontology Builder asserts an implicative copular rule containing a value comparison conditional (the second entry in Fig.4), without creating individuals linked together by the same timestamp.

At the end of the ontology building process, the agent is ready to parse a command containing the following text:



Figure 6: Inferred LODO membership after reasoning.

Give Rinazina to Robinson Crusoe

As the agent invokes the Pellet reasoner and checks for the membership of *Robinson Crusoe* to the class *Hypertensive* (as well as in Fig. 6 with Protégé), after a successful meta-reasoning of the Active Belief *eva1_sem*, the production rule 3 will match with the Beliefs KB content and the command will be discarded with an objection message from the agent. Otherwise, the production rule 4 will match and the command will be executed without any objection message. Of course, the meta-reasoning can involve also more complex queries expressed in SPARQL language and even in local closed world.

7. Conclusions and Future Work

In this paper, we presented the design of a cognitive architecture called SW-CASPAR, able to parse IoT commands from natural language and then to execute related plans. Such commands can be subordinated by a further level of (meta-)reasoning in the open-world assumption. Meta-reasoning is achieved by invoking the reasoner over an ontology representing the agent's world serialized in OWL 2, whose semantic is strictly related to the linguistic features of the idiom. The ontology is built by a semi-automatic process, taking into account the issues of natural language ontology, in order to fill the gap between expressiveness and reasoning. All the built ontologies reflect the specifications of the LODO family, which can be considered a foundational ontology aiming both at keeping the expressiveness of Davidsonian notation and at maximizing the chances of successful reasoning.

As future work, we intend to address in more depth other issues and ambiguities of natural language ontologies, in order to include additional rules in LODO, thus leading to a reasoning more human-fashioned.

References

- [1] C. F. Longo, F. Longo, C. Santoro, A reactive cognitive architecture based on natural language processing for the task of decision-making using a rich semantic, in: 21st

Workshop "From Objects to Agents" (WOA 2020), 2020.

- [2] C. F. Longo, F. Longo, C. Santoro, Caspar: Towards decision making helpers agents for iot, based on natural language and first order logic reasoning, *Engineering Applications of Artificial Intelligence* 104 (2021) 104269. URL: <https://www.sciencedirect.com/science/article/pii/S0952197621001160>. doi:<https://doi.org/10.1016/j.engappai.2021.104269>.
- [3] D. Oberle, N. Guarino, S. Staab, What is an ontology?, *Handbook on Ontologies*, 2nd edition, Springer, 2009.
- [4] L. Jean-Baptiste, Owlready2 0.31, 2021. URL: <https://pypi.org/project/Owlready2/>.
- [5] J.-B. Lamy, Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies, *Artificial Intelligence in Medicine* 80 (2017) 11–28. URL: <https://www.sciencedirect.com/science/article/pii/S0933365717300271>. doi:<https://doi.org/10.1016/j.artmed.2017.07.002>.
- [6] A. Browarnik, O. Maimon, Ontology learning from text, in: *The First International Conference on Big Data, Small Data, Linked Data and Open Data*, 2015.
- [7] F. Moltmann, Natural language ontology, 2017. URL: <https://oxfordre.com/linguistics/view/10.1093/acrefore/9780199384655.001.0001/acrefore-9780199384655-e-330>. doi:10.1093/acrefore/9780199384655.013.330.
- [8] P. van Inwagen, *Ontology, Identity, and Modality: Essays in Metaphysics*, Cambridge University Press, 2001.
- [9] M. Finkelstein-Landau, E. Morin, Extracting semantic relationships between terms: supervised vs. unsupervised methods, in: *International Workshop on Ontological Engineering on the Global Information Infrastructure*, Dagstuhl Castle, Germany, 1999.
- [10] A. S., Part-of-speech tagging and partial parsing, Young S., Bloothoof G. (eds) *Corpus-Based Methods in Language and Speech Processing*. Text, Speech and Language Technology, Springer, Dordrecht vol 2 (1997) 501–518. doi:https://doi.org/10.1007/978-94-017-1183-8_4.
- [11] P. Gamallo, M. Gonzalez, A. Agustini, G. Lopes, V. S. de Lima, Mapping syntactic dependencies onto semantic relations, in: *Proceedings of the ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering*, 2002.
- [12] D. Faure, C. Nédellec, Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system asium, *Knowledge Acquisition, Modelling and Management* (1999) 329–334.
- [13] R. Navigli, P. Velardi, Learning domain ontologies from document warehouses and dedicated websites, *Computational Linguistics*, 30(2), MIT Press (2004) 151–179.
- [14] A. A. B. Mehrnoush Shamsfard, Learning ontologies from natural language texts, *Int. J. Human-Computer Studies* 60 vol 60 (2004) 17–63.
- [15] A. Maedche, S. Staab, The text-to-onto ontology learning environment, in: *Software Demonstration at ICCS-2000-Eight International Conference on Conceptual Structures*, 2000.
- [16] S. Gillani, A. Kő, *ProMine: A Text Mining Solution for Concept Extraction and Filtering*, Springer International Publishing, Cham, 2016, pp. 59–82. doi:10.1007/978-3-319-28917-5_3.
- [17] M. R. U. Hahn, S. Schulz, *Medisyndikate—design considerations for an ontology-based*

- medical text understanding system, in: Proceedings of the AMIA Symposium (p. 330), American Medical Informatics Association, 2000.
- [18] K. Z. E. Drymonas, E. Petrakis, Unsupervised ontology acquisition from plain text: The ontogain system, *Natural Language Processing and Information System* (2010) 277–287.
 - [19] H. Y. Al-Aswadi Fatima N., Chan, K. H. Gan, Automatic ontology construction from text: a review from shallow to deep learning trend, *Artificial Intelligence Review* (2020).
 - [20] D. L. Schacter, Implicit memory: history and current status, *Journal of Experimental Psychology: Learning, Memory, and Cognition* vol. 13, 1987 (1987) 501–518.
 - [21] D. Davidson, The logical form of action sentences, in: *The logic of decision and action*, University of Pittsburg Press, 1967, p. 81–95.
 - [22] F. D’Urso, C. F. Longo, C. Santoro, Programming intelligent iot systems with a python-based declarative tool, in: *The Workshops of the 18th International Conference of the Italian Association for Artificial Intelligence*, 2019.
 - [23] C. F. Longo, C. Santoro, F. F. Santoro, Meaning Extraction in a Domotic Assistant Agent Interacting by means of Natural Language, in: *28th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*, IEEE, 2019.
 - [24] L. Fichera, F. Messina, G. Pappalardo, C. Santoro, A python framework for programming autonomous robots using a declarative approach, *Sci. Comput. Program.* 139 (2017) 36–55. URL: <https://doi.org/10.1016/j.scico.2017.01.003>. doi:10.1016/j.scico.2017.01.003.
 - [25] G. A. Miller, Wordnet: A lexical database for english, in: *Communications of the ACM* Vol. 38, No. 11: 39-41, 1995.
 - [26] World Wide Web Consortium, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004. URL: <http://www.w3.org/Submission/SWRL/>.
 - [27] H. Matthew, spacy: Industrial-strength natural language processing, <https://spacy.io>, 2017.
 - [28] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, Z. Wang, HermiT: An OWL 2 Reasoner, *Journal of Automated Reasoning* 53 (2014) 245–269.
 - [29] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, Y. Katz, Pellet: A practical OWL-DL reasoner, *Web Semantics* 5 (2007) 51–53.

ActoDemic: A Distributed Framework for Fine-Grained Spreading Modeling and Simulation in Large Scale Scenarios

Mattia Pellegrino¹, Gianfranco Lombardo¹, Monica Mordonini¹,
Michele Tomaiuolo¹, Stefano Cagnoni¹ and Agostino Poggi¹

¹*Department of Engineering and Architecture University of Parma Parma, Italy*

Abstract

Agent-based modeling and simulation techniques are widely and successfully used for analyzing complex and emergent phenomena in many research and application areas. Among the many different reasons which sustain the flexibility and success of such techniques, it is important to mention the availability of a great variety of software tools, easing (1) the development of models, (2) the execution of simulations, and (3) the analysis of results. Currently, with the rapid global spread of the COVID-19 pandemic, one of the most important research area is dedicated to define algorithms and systems to support epidemic forecasting simulations, scalable on large populations. In particular, in this paper, we propose an agent-based epidemic model and a distributed architecture that can be used for the simulation of populations represented by millions of agents. Moreover, the paper presents the results of the simulations on the data of the population of Lombardy.

Keywords

Epidemic modeling, multi-agent simulation, simulation, actor model, ABMS

1. Introduction

Spreading phenomena are widely diffused in the real world; for this reason, their modeling is crucial in several domains. For example, in biological systems, it is important to model how an infectious pathogen spreads over a population [1]; in cybersecurity, it is necessary to understand how a digital virus spreads over the nodes of a network [2]; in sociology and in Social Network Analysis, it is interesting to track how opinions and behaviors spread among a community [3]; in economics, it is interesting to study the way companies in different sectors are affected by the spreading of financial chain effects. The common point of all of these systems is that they can be represented by active entities which interact following different behaviours and models that can be generalized as a sociality factor. Different modeling techniques have been proposed to model spreading phenomena in real and complex scenarios. Two widely used techniques are System Dynamics (SD) and Agent-Based Modeling (ABM). System Dynamics analyzes the modeled system at a high abstraction level, where the interacting entities are divided into compartments.

WOA 2021: Workshop “From Objects to Agents”, September 1–3, 2021, Bologna, Italy

✉ mattia.pellegrino@unipr.it (M. Pellegrino); gianfranco.lombardo@unipr.it (G. Lombardo);
monica.mordonini@unipr.it (M. Mordonini); michele.tomaiuolo@unipr.it (M. Tomaiuolo);
stefano.cagnoni@unipr.it (S. Cagnoni); agostino.poggi@unipr.it (A. Poggi)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

A common case is the epidemiological SEIR model (Susceptible Exposed Infective Recovered) [4], where the population can move from one compartment to another according to predefined flow rates. However, the traditional SEIR model is not suitable for fine-grained modeling and not for all domains. On the other hand, agent-based approaches model the behavior of each individual agent and the interaction between agents. ABM can be used to study the system at different abstraction levels and represents an optimal choice for fine-grained simulations [5, 6].

In this paper, we propose a distributed framework (ActoDemic) that aims to facilitate the design and implementation of spreading models using Agent-Based Modeling and Simulation techniques (ABMS) for large scale scenarios. We aim to develop a general and task-independent framework. Hence, our system tries to satisfy different requirements: it has to be suitable for different domains, collaborators and computing facilities. Each agent represents an entity that is involved in some interactions in each simulation epoch, depending on its own customizable properties. Executing millions of concurrent agents could represent a bottleneck for ABMS. In order to solve this issue, we implemented the software agents as concurrent actors exploiting the ActoDeS Framework [7]. The actors have their own behavior and change it by processing asynchronous messages received from the other actors. Finally, as a use-case to test and validate our software architecture, we present a simulation of the COVID-19 outbreaks in Italy during the early-stage of the pandemic. We model about 10 millions of agents, interacting reciprocally using a social model that we have also implemented as a custom property of ActoDemic. To achieve such a result we have exploited the computational facilities of the High Performance Computing of the University of Parma, to scale and distribute the computational load of millions of agents over the available resources.

2. Literature review

The techniques to model and solve real and complex epidemic scenarios can be divided in two main sets: System dynamics techniques (SD) and agent-based modeling (ABM). System dynamics allows to analyze the modeled system at a high abstraction level where the population is divided into compartments. A common case is the SEIR model (Susceptible Exposed Infective Recovered) [4], where the population can move from one compartment to another according to predefined flow rates. However, the traditional SEIR model is not fine-grained enough to model some specific conditions, for example the lockdown policies in the case of Covid-19 management and control. This limit has motivated several research works that aim to extend that model to achieve more heterogeneity and flexibility [8, 9, 10]. However, the extensions do not address the main key-issue that is related with the main parameter, the basic reproduction number (R_0) that is not policy-invariant. Indeed, it depends on the number of interactions among the spreaders and the infection probability of the contacts. For example, it is hard to translate a real policy into the value of R_0 it will induce in the case of Covid-19 [11]. On the other hand, agent-based approaches model the behavior of each individual agent and the interaction between agents. ABM can be used to study the system at different abstractions levels. For a discussion about ABM and its advantages over system dynamics models, we refer the reader to [5] and [6]. In complex scenarios, the spreading estimation is challenging and may requires to take into account heterogeneous interaction rates among the spreaders. These

requirements can be easily modeled using ABMS. In [1] individuals are modeled as moving particles. The infections take place when two particles come closer than a certain contact radius. In [12] Social distancing for Covid-19 is modeled as changes in the contact radius or momentum equation of the particles introducing several parameters that are difficult to estimate for large scenarios. In [13], the authors model Covid-19 spreading by replacing the moving particles with contact networks for households, work and random contacts. However, to the best of our knowledge, we found a lack of models that can simulate systems with a fine-grained detail in real large scenarios. Most examples in literature are suitable for modeling a limited number of individuals and to achieve statistics that are then extended to the most general large case.

3. ActoDeS Framework

ActoDeS is a software framework that simplifies the development of concurrent and distributed systems and ensures an efficient execution of applications [7]. ActoDeS is implemented in Java and takes advantage of some implementation solutions already used in JADE [14], [15] [16], and in CODE [17]. ActoDeS has been mainly used for the development of applications in the areas of agent-based modeling and simulation [18], [19], evolutionary computation [20] and data analysis [21], [22], [23], [24], [25], [26].

ActoDeS offers a layered architecture made up of a run-time and an application layer. The run-time layer provides the software components that implement the middle-ware infrastructures that support the development of standalone and distributed applications. The application layer provides the software components that an application developer needs to extend or directly use for implementing the specific actors of an application.

In particular, an actor is an autonomous and concurrent object, characterized by a state and a behavior, that exhibits the ability to interact with other actors through the exchange of asynchronous messages. The communication between the actors is buffered: the incoming messages are stored in a mailbox until the actor is ready to process them; moreover, an actor can set a timeout for waiting for a new message and can then execute specific actions if the timeout is reached. Moreover, after the analysis of its incoming messages, an actor can send more messages to itself or to others, create new actors, update its state, change its behaviors and, finally, terminate its own execution. Each behavior can define a policy for handling incoming messages, through handlers called “cases”. Each case can only process messages corresponding to a specific pattern. Therefore, if an unexpected message arrives, then the actor mailbox maintains it until another behavior is able to process it.

As introduced above, ActoDeS can be used for developing distributed applications. In fact, depending on the complexity of the application and on the availability of computing and communication resources, an application can involve one or more computational nodes. In ActoDeS, each computational node maintains an actor space that acts as a “container” for a subset of the actors of the application and provides them with the services necessary for their execution. In particular, an actor-space contains a set of actors (application actors) that perform the tasks specific to the current application and two special actors called executor and service provider. The executor manages the concurrent execution of the actors of the actor space. The service provider enables the actors of an application to perform new kinds of actions.

4. ActoDemic

ActoDemic aims to facilitate the design and development of spreading phenomena in large-scale scenarios. The base unit is represented by actors that, depending on the target application, represent the entities of the system to model and simulate. If a fine-grained detail is required by the simulation, it is clear that a large number of concurrent actors is required too. Thus, we have used ActoDeS as a backbone to support concurrent agents, whose computational load can be distributed over several nodes. Moreover, since ActoDes is a Java-based framework, ActoDemic is able to support different operating systems and to enable fast development and prototyping, thanks to Java's built-in features such as automated serialization and extensive libraries. ActoDemic defines at least one actor space in each computational node involved in the simulation. Indeed, one of the major problems to deal with, when designing a large scale ABMS, is that the entire set of agents may not fit in a single cluster node. When a spreading phenomenon has to be modeled, the developer should define the behavior of the entities, the way they interact, and, finally, the characteristics of the spreading phenomenon. To support a wide range of spreading phenomena, ActoDemic provides a base version of the actors that can be customized by defining new behaviors in the form of new Java classes. The base actor to be used for modeling the system entities is called Base Spreader (BS). For example, if we want to model the spreading of a pathogen among people, every person can be modeled and implemented as a BS. A Base Spreader has a set of default attributes that can be enabled, configured and modified to best suit the model that is going to be realized:

- **Identification number:** Unique id that discriminates each individual.
- **Belonging to a community or cluster of entities:** As a general point in a spreading simulation, we might want to distinguish entities in different clusters and give a higher bias to the interactions among the cluster and lower outside. For example, we might want to divide entities by geographical position or define a social community. If this aspect is not required, the entities can be part of a whole cluster of entities.
- **Interaction level:** Different interaction ratios can be defined among the entities to model more complex and fine-grained scenarios;
- **Current phase:** An indicator that specifies in which contagion phase the entity is;
- **Spreading reducer:** A damper that can reduce the spreading. For example, we might want to model a vaccination, a protective device or the use of an Anti-malware in a computer network.

4.1. A Modular Software architecture

ActoDemic emphasizes software modularity for all the aspects related to the execution and management of the simulation. Moreover, a modular structure can better adapt to any proposed epidemic model and be easier to configure. The modular architecture is presented in Figure 1. The architecture is built around four modules: Agents initialization and distribution module (AID), Spreading Management (SM), Synchronization and Message passing (SMP) and, finally, a utility to generate reports and evaluate the simulation (RG).

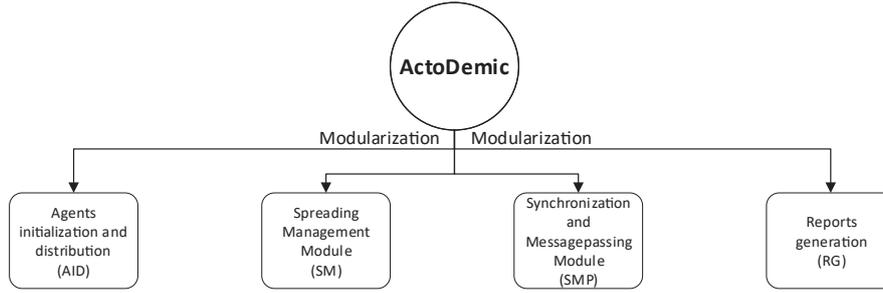


Figure 1: Framework modules

4.1.1. Agents initialization and distribution module

The AID module enables the distribution of the simulation over different nodes. It manages the initialization of all the ActoDeS entities to support the actors and the exchange of the messages. As a first step, AID initializes the required actor-spaces. Every created actor-space owns a thread and shares it with the agents that live in it. Hence, every individual is a passive actor and shares its thread with the other actors in the same space. Moreover, AID creates the base spreaders and distributes them over the nodes, according to different criteria that the developer can tune to match the system requirements:

- a. Partitioning entities according to their cluster or community, if more than the default one are defined;
- b. Splitting the population in equal-size subsets, depending on the number of actor spaces involved in the simulation.

In order to manage the base spreaders, AID module initializes the ActoDeS schedulers and managers in each actor-space presented in Section 3. The AID module is unique for each application and launches all the distributed modules. In each actor-space, each manager creates the subset of agents for its computational duties and synchronizes the simulation execution on that node with the others. Moreover, the last created manager assumes the role of “Master”. For each computational nodes it is possible to define different actor spaces. The default algorithm 1 distributes the whole population of actors over the N available actor spaces according to the identification number of each actor. This algorithm is executed by the AID module. Every subset includes, generically, the agents that go from $(n - k) \cdot p - 1$ to $(n - k + 1) \cdot p - 1$, where n is the number of partitions, k identifies the actual partition and $p = \frac{\text{population}}{N}$ is a constant.

Algorithm 1 Pseudo-code for distributing the whole agents' set across N actor-spaces

```
1:  $\mathcal{N} \leftarrow GetTotalActorSpaceNumbers()$ 
2:  $\mathcal{R} \leftarrow GetAllReferences()$ 
3: function BUILDPOPULATION( $\mathcal{N}, \mathcal{R}$ )
4:    $master \leftarrow amITheMaster()$ 
5:    $Begin \leftarrow \emptyset$ 
6:    $End \leftarrow \emptyset$ 
7:   if  $master$  is True then
8:     for ( $i = 0; i < \mathcal{N}; i ++$ ) do
9:        $\mathcal{S} \leftarrow ((\mathcal{N} - i) \cdot p - 1)$ 
10:       $\mathcal{E} \leftarrow ((\mathcal{N} - i + 1) \cdot p - 1)$ 
11:       $SendMessage(\mathcal{R}[i], (\mathcal{S}, \mathcal{E}))$ 
12:    end for
13:  end if
14:  loop
15:    Wait a message from the master Actor-space ▷
16:  end loop
17:   $Begin, End \leftarrow ProcessMessageFromMaster()$ 
18:   $CreatePopulation(Begin, End)$ 
19: end function
```

where:

- \mathcal{N} is the number of total Actor-Spaces that we want to create;
- \mathcal{R} is a set that contains all the Actor-Space references (unique system-wide id that we need to reach an actor and communicate with it);
- $master$ is a Boolean value that specifies whether an Actor-space acts as a master or not;
- $SendMessage()$ is a function that communicates to an Actor-space which population subset it has to manage.
- $Begin, End$ define the range of the actors to be created and managed ;
- $CreatePopulation()$ divides the population into subsets

Algorithm 2 Pseudo-code to distribute X actor-space across N computational nodes

```
1:  $\mathcal{N} \leftarrow GetNodesNumber()$ 
2:  $\mathcal{T} \leftarrow GetTaskPerNodesNumber()$ 
3: function SPREADACTORSPACES( $\mathcal{N}, \mathcal{T}$ )
4:    $N\_JOB \leftarrow \mathcal{N} \cdot \mathcal{T}$ 
5:    $BrokerIp \leftarrow \emptyset$ 
6:   for ( $i = 0; i < N\_JOB; i ++$ ) do
7:
8:     if  $i == 0$  then
9:        $BrokerIp \leftarrow SetBrokerIp()$ 
10:       $LaunchActoDemic(broker, BrokerIp)$ 
11:    else if  $i == (N\_JOB - 1)$  then
12:       $LaunchActoDemic(initiator, BrokerIp)$ 
13:    else
14:       $LaunchActoDemic(node, BrokerIp)$ 
15:    end if
16:  end for ▷ The For cycle is managed by MPI on SLURM
17: end function
```

where:

- \mathcal{N} : Number of computational nodes;
- \mathcal{T} : Number of tasks using a single CPU on every node;
- N_JOB Number of actor-spaces that will be created;
- *LaunchActoDemic()* is the function that launches an ActoDemic instance and it takes two arguments.

We can distinguish three entities that play a fundamental role in ActoDemic and are inherited from ActoDeS: the Broker, the generic node and the Initiator. The Broker receives messages from producers and sends messages to consumers. The Initiator acts like the master node and coordinates the agent creation process across the various actor-spaces. AID provides also a sub-module that enables the use of ActoDemic with the Slurm Workload Manager (SLURM) and the MPI protocol [27]. SLURM is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters. The MPI protocol is useful to distribute and manage the actor-spaces across the computational nodes. These capabilities enable the use of ActoDemic in High Performance Computing scenarios where SLURM and MPI are often a standard. An explanation of this sub-module is given by algorithm 2.

4.1.2. Spreading Management Module

ActoDemic exploits some concepts from Network Science to model the interactions among the system entities. The Spreading Management module (SM) defines the way the interactions should occur. Every actor-space manages its own actors, as well as their own different interactions. It assumes that a generic actor represents a node in a generic graph and the outgoing and incoming links represent, respectively, the entities with which the node interacts and vice versa. Graph theory allows us to study the distribution of interactions, by studying the node degree distribution. The distribution of interactions is a crucial factor that affects the spreading of an epidemic phenomenon. ActoDemic provides a simple way for tuning this distribution when defining the interactions. It allows the user to choose a distribution and set it as default. The available distributions that the framework provides are: power-law, log-normal, exponential, Gaussian and Poisson. The SM module also allows to partition the entire agents' set into different clusters and give a bias to the interactions such that they are higher inside the cluster and lower outside.

When two individuals interact with each other, a procedure that simulates an infection is started. We implemented an epidemic diffusion model starting from the compartments of the SEIR mathematical model (Susceptible-Exposed-Infective-Recovered)[4]. SEIR is based on a series of dynamic differential equations that consider the amount of the population subject to contagion, the trend over time of the number of individuals who recover after infection, and of the casualties. A limit of this model is its coarse-grain nature with respect to individual behaviors. Moreover, the SM module provides a method to control the contagion power through a variable called "Transmission Probability" (TP). This variable can take values between 0 and 1, to express the probability for a contagion operation between two entities to be successful.

However, the compartments are fully customizable. A single compartment can be disabled, changing in this way the development of the infection. Moreover, intermediate compartments can also be added to add steps to the simulation process and even the duration between the various phases can be tuned. ActoDemic also makes it possible to enable and specify the behavior in case of a re-infection event after a recovery.

An example of a full infection cycle is shown in Figure 2. This scheme represents the COVID-19 infection cycle. We shall deepen this aspect in section 5.

Finally, the SM module is responsible for the initialization and the implementation of the damper that is able to scale the transmission probability.

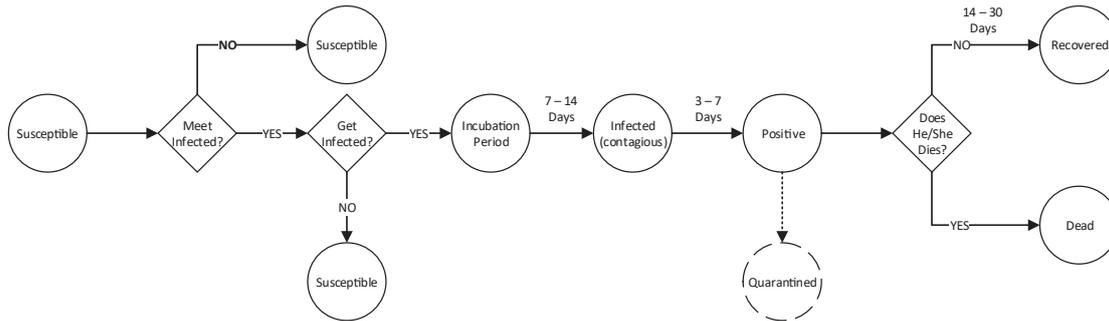


Figure 2: Example of a generic infection Cycle

4.1.3. Synchronization and Message passing Module

The SMP module manages the message exchange and time synchronization.

ActoDemic uses a very simple scheduler called "*CycleScheduler*" provided by ActoDeS. This tool can be used in a wide variety of applications; more specifically, also in ABMS applications. Furthermore, this scheduler manages the passive actors within its actor-space and cyclically repeats the same actions, until the simulation ends:

1. Send a "step" message to all agents and increment the "step" value; this operation triggers the transition from one epoch to the next.
2. Perform an execution step of all agents.

Every actor-space is modeled as an actor and thus an actor cannot access the internal state of other actors. Therefore, if one or more Base Spreader that belongs to an actor-space interact with some other Base Spreaders of a different actor-space then we must inform both actor-spaces of that interaction. This can only be done with a messages exchange system.

Another typical property of the actor model is that the exchange of messages is asynchronous. This property creates a synchronization problem, because an actor-space could move to the next epoch without receiving all the necessary information from all the other actor-spaces. Moreover, an actor-space needs to retrieve the information from all other actor-spaces before triggering the transaction of its actors to the next epoch. SMP solves this problem by creating a mechanism that acts as a "barrier" and does not allow all the actor-spaces to switch to the next epoch if the message exchange between all the actor-spaces has not been completed.

Information routing and management may differ according to the target model the user wants to implement. However, there is a basic scheme that we report here. Some information is sent across every partition type, while some information can only be sent or received by the master. The content of a generic message can be schematized in this way:

1. Information sent to all partitions:
 - Information about Base Spreaders' interactions: it informs the actor-space of the interaction between two or more of its Base Spreaders;
 - Base Spreaders that have to change their status: it specifies which Base Spreaders have to change their status due a trigger situation;
 - Base Spreaders that have been closed: actors that have to end their cycle;
 - Statistical Data: data needed to generate reports;
 - Synchronization message: message that acts like a “barrier”. When an actor-space receives all the synchronization messages from all the actor-spaces involved in the simulation, then it can move to the next epoch;
 - End signal: a message that allows the end of simulation process;
2. Information sent to master only:
 - Partial summary report: data needed to generate the final reports;
3. Information sent only by the master:
 - Base spreaders that have to shutdown themselves: the master decides who should end their execution, based on the user's criteria.

4.1.4. Reports generation module

The RG module enables the generation of reports during the simulation process. The user can decide whether to enable, or not, the generation of reports. Moreover, the user can decide how often a report is to be generated.

In general, at the end of every epoch every actor-space generates a report that regards only its partition and its actors: these are called “intermediate reports”. Furthermore, the intermediate reports are also used to support a “Save&Load” functionality. Thanks to this mechanism it is possible to restart the simulation process from a specific epoch. However, in order to do so, reports need to include certain information for each Base Spreader within the actor-space:

- The Identification Number
- The belonging to any type of social cluster
- The interaction level
- The epidemic phase it is currently going through
- How much the epidemic phases must last
- If the Base Spreader has an active Spreading reducer
- A set that maintains the information about the past interactions with the other Base Spreaders

Finally, at the end of the simulation, the master node generates a final report that summarizes the most important and relevant information. In particular, it generates a summary for each simulation epoch showing how many people belong to each specific epidemic compartment.

The RG module is also the one responsible for the initial configuration. Thanks to this, ActoDemic supports an external configuration file, in which it is possible to customize and specify all the properties explained in the article.

5. Use-Case: COVID-19 spreading in Lombardy

5.1. Modelling

After building and modeling the framework in all its features, we needed a real use case to test it. Therefore, to validate ActoDemic, we focused our attention on COVID-19 spreading. The social interactions, which are the major cause of COVID-19 spreading, can be easily modeled by properly setting the parameters on which ActoDemic depends.

The proposed model simulates about ten million independent agents that reproduce the social behaviour of the inhabitants of Lombardy, a region in northern Italy. We have decided to simulate the COVID-19 epidemic spread in Lombardy for several reasons. It was the first region, in Italy, to be affected by the virus and currently it is the first for number of infections; therefore, it is the Italian region that offers the most abundant statistical data, that can be used to compare and validate our model. This represents an interesting use-case to test the robustness of our methodology.

To make ActoDemic suitable for our use-case, we have customized every framework module. Every person involved in the simulation process is represented by a Base Spreader and every epoch represents a generic day in a real-life situation.

The ten million people living in Lombardy are subjected to a partition by districts. Accordingly, we have divided the entire population geographically respecting the number of Lombardy provinces and the distribution of their inhabitants, creating several social communities. We have assigned an additional feature to each Base Spreader: its age. The alter parameter is crucial, because it adds information in our design and can be used to better model the social interactions. We have also respected the Lombardy age distribution [28]. To model social interactions even more efficiently, we have defined three interaction ratios: high, medium and low. These have been introduced to increase or decrease the average number of the subjects' daily contacts with other people, based on their age. To estimate the average number of a Base Spreader's contacts, we have used data from the Italian National Institute of Health [29] and [30].

To correctly model the pathogen spread, we have customized the SM module. We have used the base SEIR model, adding two extra compartments: Positive and Quarantine. These phases are typical in the COVID-19 infection cycle. Initially, all people are in the susceptibility stage. In this compartment, every subject can be infected by another one who is contagious. An individual who is infected moves from a susceptibility phase to an incubation phase and remains in this stage for a certain time, before moving into an infection stage. A subject in this condition can infect other people. When this phase ends, the person becomes positive. After a certain time, a positive will either heal or die. There is no death probability, but deaths follow the real death curve trend in Lombardy. When an individual heals, it cannot be infected any more. In particular, the incubation phase lasts from 7 to 14 days, the infectious phase from 3 to 7 days, and the positive phase from 14 to 30 days [31] [32]. Figure 2 shows a diagram that represents the infection cycle.

To model COVID-19 compartments we have used preliminary data collected by [31], [32] and the age susceptibility to COVID-19 virus. Moreover, the SM module supports a damper to mitigate the contagion spread; we have used this particular feature to simulate the adoption of protective devices. We have collected data about the percentage of the population that was using protective devices [33] and their effectiveness [34]. The last property expected by the SM module concerns the distribution of social interactions. For this reason, we have evaluated various hypotheses, but, in the end, we have decided to focus our studies on a power-law distribution. We assume a common hypothesis in network science that asserts that social networks commonly have a power-law distribution with an exponent between 2 and 3, also known as the scale-free property [2]. Contact networks are usually modeled with a power-law distribution [35]. We have exploited these interaction ratios to also model the lockdown policy and the contagion containment strategy adopted in Italy in the first pandemic months. Modeling the Italian lockdown has required different pieces of information about the set of “essential workers” [36]. Remember that the only people who were not subjected to limitations were those who worked in the so-called "essential sectors".

The information that the actor-spaces exchange with each other need to be modified to make the simulator work. Accordingly, we have customized the SMP module to modify the message contents:

1. Information sent to all partitions:
 - Information about people’s meeting
 - People who have to change their infection phase to "Incubated" due an infection
 - Number of people expected to die in that partition
 - Statistical Data
 - Synchronization message
 - End signal
2. Information sent to master only:
 - Currently positive people
 - Currently infected people
 - Partial summary report
3. Information sent only by the master:
 - Total people expected to die

The last module we have customized is the RG module. In our use-case, we generate an intermediate report at the end of every epoch. Obviously, the information included in each intermediate report has been customized ad-hoc for our use-case. It contains the following information: id, age, province of residence, essential worker (Boolean value), mask wearing (Boolean value), incubation days period, infection days period, positive days period and usual Contacts’ List.

At the end of the simulation process, a summary report is generated, containing the following information: epoch, positive people, infected people, infected people, susceptible people, recovered people, dead People.

Once every customization has been realized, the framework is ready to be tested and to reproduce some results. In the next section, we report the results we have obtained in our use-case.

5.2. Experimentation

In order to evaluate our simulator, we have considered two different COVID-19 outbreaks in Lombardy (Italy). In particular, we are interested in modeling the first wave from January to April 2020 and the second one between August and December 2020. The combination of these two waves has been taken in consideration to validate our simulator and for modelling its parameters.

When the simulation process starts, all agents are in the susceptibility status, as previously reported. In this way, no one can start a hypothetical contagion. Hence, at the beginning of the simulation we have to choose randomly which Base Spreader will start directly from an incubation phase. In addition, to do this, we respect the number of positives people between 20 and 29 February in Lombardy on a provincial basis.

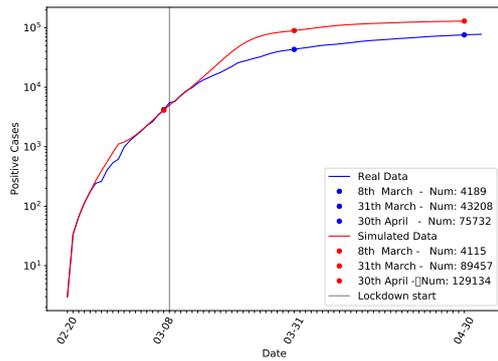
We have calibrated the transmission probability with a random-search over the probability space. We have estimated this value trying to chase up the contagion curve until the pre-lockdown date, March 8th, 2020. The lockdown is an Italian policy to prevent the contagion spread that implies the closure of non-essential activities, social distancing and some rules for limiting the movement of people. The value satisfying these hypothesis is 0.3. A summary histogram is shown in figure 3b.

5.3. Results

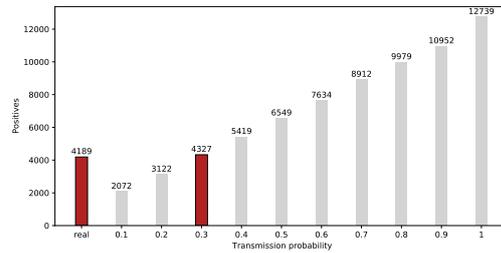
In this section, we present the results that we have obtained simulating different scenarios and considering each time an average of 10 different runs, since the entire simulation process is stochastic in most of its steps. For each case, we have measured the simulation quality using the Pearson correlation and the Root-mean-square error (RMSE) between simulated data and real data from the Italian government [37]. The Pearson correlation expresses any linear relationship between two statistical variables. This value ranges from -1 to 1 , where 1 corresponds to a strong positive linear correlation and -1 corresponds to a strong negative linear correlation. In our case, it explains how much the trend of the simulated contagion curve resembles the real one. The Root-mean-square error is computed between the predicted values and the real data. Figure 3a shows the results of the simulation in the early-stage of the pandemic between January and April 2020 with the COVID-19 Transmission probability (CTP) equal to 0.3. The blue curve represents the real contagion data, while the red curve represents the simulated data. Pearson correlation and RMSE referred to Figure 3a until April 30th are equal to 0.992 for Pearson correlation and 38818, respectively. The number of the total positives obtained using the simulator in that date, exceeds by about 53,000 units the number of actual positives (Figure 3a).

The results in Figure 3a seem to support our thesis, but it is widely conceivable that the real data measured over that period were underestimated.

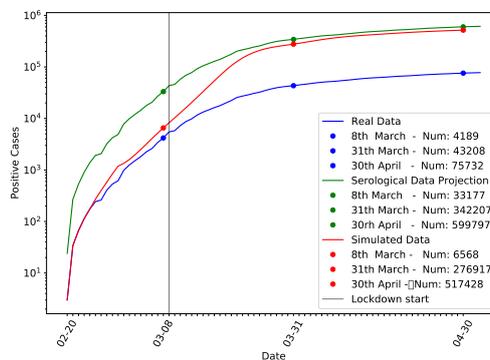
A comparison with the ISTAT's serological investigation reveals a very different situation [38]. This study shows that, on July 15th, the actual number of COVID-19 cases in Lombardy was about 7.92 time greater than the data from COVID-19 tests. In addition, the study shows that about 7.5% of the Lombard population had developed antibodies for the COVID-19. The population of Lombardy is about 10,060,000 people, 7.5% of which is therefore equivalent to



(a) Simulation results with real data with starting assumptions (CTP=0.3) - Incremental positives representation



(b) Positive people on 8th March 2020 with different transmission rates



(c) Simulation results with real data with transmission probability equal to 0.53 - Spring case - Incremental positives representation

about 754,500. This strengthens the hypothesis that the spring data were underestimated. Assuming that this ratio is constant over time, we retro-projected this data and observed how many positive people could be estimated.

Therefore, we have used the data of the second wave (August to December 2020) to perform a fine tuning procedure. We have used the previously obtained CTP to verify whether the simulated data properly followed the real data generated by the second wave. The values obtained greatly underestimated the actual data. For this case, the Pearson correlation is 0.988 and the RMSE is 27415.

Due to the large estimation error, we have decided to estimate again the transmission probability parameter, using the data of the second wave. We have searched for a value that follows the contagion curve correctly. The best value that satisfies our hypothesis is 0.53. In this final case the Pearson correlation is 0.996 while the RMSE is 6405.

5.4. Final Projections

In the light of the previous considerations, we have decided to simulate again the first wave with the new CTP value. Moreover, we have added also the data obtained from the comparison with the national screening activity. The whole process is shown in Figure 3c. The blue curve represents the actual data, the green curve represents the serological data projection on the real data and the red curve represents the simulated data with transmission probability equal to 0.53. The second estimation of the COVID-19 Transmission Probability using the autumn data is confirmed as a better choice to validate our model. The difference with the serological data projection on April 30th, corresponding to the last simulation day, is only 83,369 units. Seroprevalence analysis is much more reliable than the data collected during the months of March and April, because it also takes into account asymptomatic people, which is a very crucial factor. Considering the cumulative curves in Figure 3c, the Pearson correlation is 0.996 and the RMSE is 249,529. In the comparison between simulated and serological data Pearson correlation is again 0.996 while the RMSE is 56,009. Matching data confirms the validity of our hypothesis and of our simulation model.

6. Conclusion

In this paper, we have presented a framework that aims to combine a fine-grained spreading model with a large-scale scenario. This result is achieved by exploiting an efficient multi-agent system that can be run on a distributed architecture. The proposed framework has been designed with a modular structure to be user-friendly and easy to customize. We have validated our framework simulating the outbreaks of COVID-19 in Lombardy (Italy) in 2020.

Actodemic met our expectations in our use-case. Indeed, simulating 10 millions concurrent agents requires several resources around 600-800 GB of Ram, 32 CPUs with an average execution time for each simulation equal to 6 hours. Obviously these results have been possible also thanks to the use of the High Performance Computing facilities. The results indicates that the framework is able to simulate and accurately reproduce a spreading phenomenon. The experiment provides a new insight into the spreading modelling with a multi agent system, which allows to model the agents' behaviour at a low abstraction level.

Future developments are related to establish how much the framework can become better by adding more features and modules to make ActoDemic faster and more customizable and to find a way to implement and support a higher number of agents with less resources. Moreover, different solutions for message exchanging and other multi-agent paradigms could lead to improvements that would make the framework even more performing.

References

- [1] P. C. Silva, P. V. Batista, H. S. Lima, M. A. Alves, F. G. Guimarães, R. C. Silva, Covid-abs: An agent-based model of covid-19 epidemic to simulate health and economic effects of social distancing interventions, *Chaos, Solitons & Fractals* 139 (2020) 110088.
- [2] A.-L. Barabási, *Network Science*, Cambridge University Press, 2016.

- [3] G. Lombardo, P. Fornacciari, M. Mordonini, L. Sani, M. Tomaiuolo, A combined approach for the analysis of support groups on facebook-the case of patients of hidradenitis suppurativa, *Multimedia Tools and Applications* 78 (2019) 3321–3339.
- [4] H. W. Hethcote, The mathematics of infectious diseases, *SIAM review* 42 (2000) 599–653.
- [5] G. P. Figueredo, P.-O. Siebers, M. R. Owen, J. Reys, U. Aickelin, Comparing stochastic differential equations and agent-based modelling and simulation for early-stage cancer, *PloS one* 9 (2014) e95150.
- [6] H. V. D. Parunak, R. Savit, R. L. Riolo, Agent-based modeling vs. equation-based modeling: A case study and users’ guide, in: *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, Springer, 1998, pp. 10–25.
- [7] F. Bergenti, E. Iotti, A. Poggi, M. Tomaiuolo, Concurrent and distributed applications with actodes, in: *MATEC Web of Conferences*, volume 76, EDP Sciences, 2016, p. 04043.
- [8] D. Acemoglu, V. Chernozhukov, I. Werning, M. D. Whinston, Optimal targeted lockdowns in a multi-group SIR model, volume 27102, National Bureau of Economic Research, 2020.
- [9] J. R. Donsimoni, R. Glawion, B. Plachter, K. Wälde, Projecting the spread of covid19 for germany (2020).
- [10] V. Grimm, F. Mengel, M. Schmidt, Extensions of the seir model for the analysis of tailored social distancing and tracing approaches to cope with covid-19, *medRxiv* (2020).
- [11] C. Avery, W. Bossert, A. Clark, G. Ellison, S. F. Ellison, An economist’s guide to epidemiology models of infectious disease, *Journal of Economic Perspectives* 34 (2020) 79–104.
- [12] E. Cuevas, An agent-based model to evaluate the covid-19 transmission risks in facilities, *Computers in biology and medicine* 121 (2020) 103827.
- [13] R. Hinch, W. J. Probert, A. Nurtay, M. Kendall, C. Wymatt, M. Hall, K. Lythgoe, A. B. Cruz, L. Zhao, A. Stewart, et al., Openabm-covid19-an agent-based model for non-pharmaceutical interventions against covid-19 including contact tracing, *medRxiv* (2020).
- [14] A. Poggi, M. Tomaiuolo, P. Turci, Extending jade for agent grid applications, in: *13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, IEEE, 2004, pp. 352–357.
- [15] A. Negri, A. Poggi, M. Tomaiuolo, P. Turci, Dynamic grid tasks composition and distribution through agents, *Concurrency and Computation: Practice and Experience* 18 (2006) 875–885.
- [16] F. Bergenti, G. Caire, S. Monica, A. Poggi, The first twenty years of agent-based software development with jade (2020).
- [17] F. Bergenti, A. Poggi, M. Tomaiuolo, An actor based software framework for scalable applications, *Lecture Notes in Computer Science* 8729 (2014) 26–35.
- [18] A. Poggi, Agent based modeling and simulation with actomos., in: *WOA*, 2015, pp. 91–96.
- [19] G. Angiani, P. Fornacciari, G. Lombardo, A. Poggi, M. Tomaiuolo, Actors based agent modelling and simulation, in: *International Conference on Practical Applications of Agents and Multi-Agent Systems*, Springer, 2018, pp. 443–455.
- [20] G. Petrosino, F. Bergenti, G. Lombardo, M. Mordonini, A. Poggi, M. Tomaiuolo, S. Cagnoni, Island model in actodata: an actor-based implementation of a classical distributed evolutionary computation paradigm, in: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2021, pp. 1801–1808.
- [21] G. Lombardo, P. Fornacciari, M. Mordonini, M. Tomaiuolo, A. Poggi, A multi-agent architecture for data analysis, *Future Internet* 11 (2019) 49.

- [22] E. Franchi, A. Poggi, M. Tomaiuolo, Blogracy: A peer-to-peer social network, in: *Censorship, Surveillance, and Privacy: Concepts, Methodologies, Tools, and Applications*, IGI global, 2019, pp. 675–696.
- [23] E. Franchi, A. Poggi, M. Tomaiuolo, Social media for online collaboration in firms and organizations, in: *Information Diffusion Management and Knowledge Sharing: Breakthroughs in Research and Practice*, IGI Global, 2020, pp. 473–489.
- [24] M. Tomaiuolo, G. Lombardo, M. Mordonini, S. Cagnoni, A. Poggi, A survey on troll detection, *Future internet* 12 (2020) 31.
- [25] G. Lombardo, A. Poggi, Actornode2vec: An actor-based solution for node embedding over large networks, *Intelligenza Artificiale* 14 (2020) 77–88.
- [26] G. Lombardo, A. Poggi, A scalable and distributed actor-based version of the node2vec algorithm., in: *WOA*, 2019, pp. 134–141.
- [27] L. Clarke, I. Glendinning, R. Hempel, The MPI message passing interface standard, in: *Programming Environments for Massively Parallel Distributed Systems*, Birkhäuser Basel, 1994, pp. 213–218. URL: https://doi.org/10.1007/978-3-0348-8534-8_21. doi:10.1007/978-3-0348-8534-8_21.
- [28] Popolazione per età, sesso e stato civile 2020, <https://www.tuttitalia.it/lombardia/statistiche/popolazione-eta-sesso-stato-civile-2020/>, 2020.
- [29] C. T. Scientifico, Valutazione di politiche di riapertura utilizzando contatti sociali e rischio di esposizione professionale (2020).
- [30] J. Mossong, N. Hens, M. Jit, P. Beutels, K. Auranen, R. Mikolajczyk, M. Massari, S. Salmaso, G. S. Tomba, J. Wallinga, et al., Social contacts and mixing patterns relevant to the spread of infectious diseases, *PLoS Med* 5 (2008) e74.
- [31] Covid-19: nuovo studio sull'incubazione del sars-cov-2, <https://www.osservatoriomalattie.it/news/ricerca-scientifica/15771-covid-19-nuovo-studio-sull-incubazione-del-sars-cov-2>, 2020.
- [32] Epidemia covid-19, https://www.epicentro.iss.it/coronavirus/bollettino/Bollettino-sorveglianza-integrata-COVID-19_12-marzo-2020.pdf, 2020.
- [33] D. Perrotta, A. Grow, F. Rampazzo, J. Cimentada, E. D. Fava, S. Gil-Clavel, E. Zagheni, Behaviours and attitudes in response to the COVID-19 pandemic: Insights from a cross-national facebook survey (2020). URL: <https://doi.org/10.1101/2020.05.09.20096388>. doi:10.1101/2020.05.09.20096388.
- [34] S. E. Eikenberry, M. Mancuso, E. Iboi, T. Phan, K. Eikenberry, Y. Kuang, E. Kostelich, A. B. Gumel, To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic, *Infectious Disease Modelling* 5 (2020) 293–308. URL: <https://doi.org/10.1016/j.idm.2020.04.001>. doi:10.1016/j.idm.2020.04.001.
- [35] S. Eubank, H. Guclu, V. A. Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, N. Wang, Modelling disease outbreaks in realistic urban social networks, *Nature* 429 (2004) 180–184.
- [36] Coronavirus, la demografia del lockdown: ecco l'identikit di chi lavora e di chi sta a casa, <https://www.ilfattoquotidiano.it/2020/04/13/coronavirus-la-demografia-del-lockdown-ecco-lidentikit-di-chi-lavora-e-di-chi-sta-a-casa/5763773/>, 2020.
- [37] Github protezione civile, <https://github.com/pcm-dpc/COVID-19>, Real Time.
- [38] Studio della sieroprevalenza in italia, http://www.salute.gov.it/portale/news/p3_2_1_1_1.jsp?lingua=italiano&menu=notizie&p=dalministero&id=4998, 2020.

Risk Sensitivity of Production Studios on the US Movie Market: an Agent-based Simulation

Francesco Bertolotti¹, Sabin Roman²

¹Università Carlo Cattaneo – LIUC, Corso G. Matteotti, 22, Castellanza (VA), 21053, Italy

²Centre for the Study of Existential Risk, University of Cambridge

Abstract

The movie industry is a highly product differentiated industry where firms mainly compete in non-price product attributes. The success of a movie on the film distribution market depends on a variety of factors. Because of the short life cycle, the rapid decay in revenues, and the constant entrance of new competitive products, temporal decisions play a crucial role. The time series of the number of movies on release and the sum of the box office results of the ten top movies (ranked by box office result for that week) show that a seasonality emerges in the US movie market. Moreover, the two time series are on counterphase. We suggest the reason is a risk sensitivity adaptation in the behaviour of the movie's distributors. This paper tests this hypothesis. We develop an agent-based model of a movie market, and we simulated it for 15 years. We show that a comparable global behaviour exists when producers schedule the movies according to given risk-sensitive strategies. Our analysis improves the knowledge of the US motion picture market and may support film producers on how to change their scheduling decisions.

Keywords

movie market, agent-based-modelling, box office, risk sensitivity, risk preferences

1. Introduction

The movie industry is a peculiar industry in which a small number of companies compete with each other to get the attention of a fixed number of customers. Moreover, the motion picture is a unique product that can not be differentiated by price. The research on the area for a long time has concentrated on understanding the factors that influence the box office success of a film [1, 2] as a way to address the high risk related to the movie industry. Besides, the combination of these factors makes the competition landscape extremely uncertain [3]. The movie box office industry is an 11.4\$ billion a year business only in the North American market [4]. This economic importance has a dark side: the entity of the budget necessary to produce a successful motion picture. That is why dealing with the risk on the film market is so important. Such as other entities [5, 6], production studies deal with uncertainty developing and adapting their risk preferences. Consequently, the risk-sensible actions of individuals could affect the global behaviour of the system. Figure 1 shows the time series of the normalized number of

WOA 2021: Workshop "From Objects to Agents", September 1–3, 2021, Bologna, Italy

✉ fbertolotti@liuc.it (F. Bertolotti); sr911@cam.ac.uk (S. Roman)

🆔 0000-0003-1274-9628 (F. Bertolotti); 0000-0002-7513-3479 (S. Roman)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

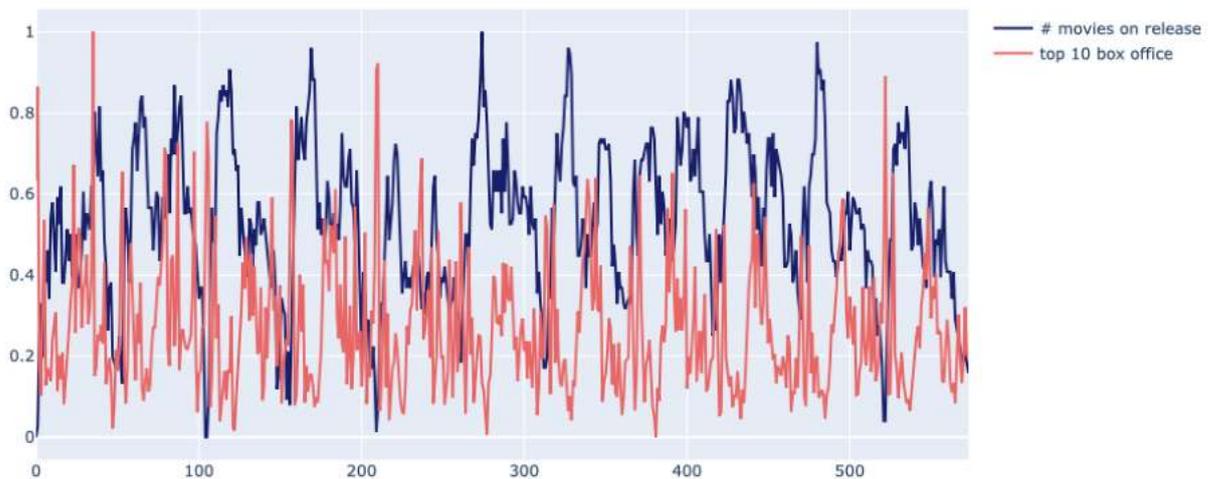


Figure 1: Time series of the number of movies on release each week and of the box office results of the top 10 ranking movies for box office on the US movie market (both values are normalized)

movies on release each week and the box office results of the top 10 ranking movies (ranked for box office) on the US movie market.

It is observable that the two time series are in counterphase and repetitively intersect with a seasonal trend. We suggest that it derives from the risk-sensible behaviour developed by the producer studios to address uncertainty. This paper aims at confirming this hypothesis, showing how production companies in the US movie market adapt their risk attitudes. To reach this goal, we follow a three-step process. First, we develop an agent-based model (ABM) of the US movie market. In the model, agents (the production studios) decide regarding the production and scheduling of movies. The model simulates the competition dynamics and the effect of different scheduling risk strategies. The calibration consists of two phases. The initialization of producers draws from real-world data collected from a novel database, except risk sensitivity. Later, we simulate the model and calibrate the global behaviour on the time series shown in Figure 1, adjusting the risk preferences of agents. Finally, the resulting risk sensitivities of the producers' agent are compared with the other producers' features. The analysis highlights a direct relationship between the producers' investments and their risk aversion and an inverse relationship between the amount of budget invested by a company and the variability of the risk aversion.

The rest of the paper proceeds as follows. In Section 2, an overview of the research about the movie market is provided. Section 3 shows the model employed for this research, while Section 4 describes the calibration methodology. The main results of this research are deployed in Section 5. Finally, Section 6 presents the conclusions.

Table 1

Main research on the movie market that employs agent-based modelling

Source	Brief description
De Vany and Lee 2001	Observe effect of word of mouth on the box office revenues
Delre et al. 2007	Explore the social influences on movie market inequalities
Broekhuizen et al. 2011	Explore the social influences on movie market inequalities in different
Delre et al. 2017	Analyze the dynamics of competition between two production studios
Iasello 2017	Explain the racial minority underrepresentation in Hollywood movies
Satoh and Matsubara 2021	Predict the box office result

2. Background

Research around the movie market is a field that has more than fifty years of history [7]. Hence, it is not in the scope of this paper to review the literature on the topic. We only provide an overview of the specific issues and sub-areas affected by our study.

One of the more popular themes is the understanding of which factors influence the box office results. Literature addressed this issue in various ways. This is a consistent interest in the influence of sociality on the box-office results. It can affect the power of critics on the box office results [8] as well as the effects of social interactions, both physical [9] and virtual [10]. Similarly, the sentiment of word of mouth and internet reviews seems to influence the potential attendance success of a motion picture [11, 12]. In these papers, usually, a large set of independent pre-release variables are employed to investigate how they affect the future success of a movie, such as production budget, critic rating, MPAA rating, star power, and genre [1, 2]. Recently, the box office success of a motion picture is forecast employing new predictive technologies. Neural networks and other artificial intelligence techniques proved to be particularly effective on this task [13, 14, 15]. Also, big data analysis on specific kinds of interactions has the potential to improve the forecasting power on the box office result of a movie [16].

Nonetheless, these researches tend to under-evaluate the presence of a complex and uncertain environment and the importance of competition. De Vany and Walls first investigated these subjects, focusing on a possible strategy (the inclusion of more stars in the cast of the movie) to reduce the risk of the box office [3]. They concluded that there did not exist any viable strategy to eliminate the uncertainty because it is not possible to appraise the causal effect of each factor on the success of a movie. Analogously, Ribera and Sieber 2009 and Von Rimscha 2009 focused on the different managerial strategies that production studios should follow to address the uncertainty [17] [18], while Bi and Giles concentrated on the development of a measure to define risk and expected shortfall on a movie market [19].

Regarding competition, papers focused on the positioning of the positioning to perform a

good box office result, such as debut at number 1 [20] or avoiding to fail early surviving enough weeks on the market [21]. Gutierrez-Navratil et al., in two consecutive papers, address the undirect interaction of different producers strategies [22], arguing that, if not colluding, major distribution studios achieved a significant rate of coordination on the release scheduling [23].

From a methodological perspective, we identified six studies [24, 25, 26, 27, 20, 28] which employed agent-based modelling to study the movie market, especially to take into account the role of low-level interaction on the global output (which in general is the box office result of movies). Table 1 resumes the main findings.

Nevertheless, as stated at the beginning of the section, this is just an overview of the topic. More comprehensive reviews of the literature were recently published [29, 30].

3. Agent-based model

This model simulates the competition dynamics between production studios in the US movie market. The purpose is to understand how its global behaviour derives from decision makers' risk preferences. Hence, the model focuses on the movie production studios (from now on "producers", for simplicity). We employ agent-based modelling because it is well-suited to simulate individual behaviours and appraise their effect on the overall system [31]. This model contains two kinds of entities: movies and producers. Each simulation runs for 780 time-steps, which stands for 15 years divided into time steps of one week.

Movies are passive objects and can be created, scheduled, released and retired by the producers. Each movie owns five main features:

1. owner: the producer agent that creates the movie for the first time.
2. quality: the goodness, that is the share of the success of a movie not addressed to its production budget.
3. budget: the number of dollars that a producer invested in the creation of the movies.
4. weeks needed to completion: the number of time steps a producer necessitates to develop a movie. It depends linearly on the budget.
5. potential market: the number of spectators that want to see the movie in a theatre. For simplicity, we suppose that each movie could be seen only one time by each spectator. So, when a given number of spectators attend the movie, the potential market of the movie diminishes the same number for the following week.

The "producers" agents stand for the production and distribution companies that compete in the US movie market. The modelling of producer agents follows some assumptions. They tend to maximize the profit and not to cooperate with other agents. Besides, each producer knows the other producers (as well as their scheduling activities). The last point implies that whenever a producer schedules the release of a movie, the other producers know its scheduled release date and the budget (but not its quality). This information process acknowledges both the business intelligence activities of movies firms and the presence of non-perfect information

(e.g., the producers do not know in advance the quality of movies scheduled by other producers). Besides, the information related to the competitive landscape is computed in the same way by each agent. Heterogeneity relates to how this knowledge is employed to make scheduling decisions. Hence, the model accounts for competition by indirectly connecting heterogeneous agents of the same kind.

Four features characterize producers:

1. mean budget: the mean budget of the movies develops by a specific producer.
2. budget distribution: the distribution of the budgets of the movies developed by a specific producer. The analysis of real data permits the identification of three kinds of feasible distributions for the budget decision of each producer: power law, gamma, uniform.
3. frequency of release: the mean number of new releases scheduled for each new week.
4. risk sensitivity: preferences related to the competition during the scheduling decision-making. Agents with high-risk sensitivity try to avoid competition (e.g., risk-averse), while agents with low-risk sensitivity schedule their movies with other movies and seek competition (e.g., risk-seeking).

Producers decide about the creation, the completion, the scheduling and the retiring of a movie.

The decisions related to creating new movies are taken by computing a Poisson random variable, which lambda is the frequency of release of the producers. The creation activity generates the budget and the quality. The budget is sampled from the distribution of the budgets of the producer. The quality of a new movie is independent of the producers and sampled from a continuous uniform distribution between 0 and 1.

Producers work on a movie when its rate of completion is below 100%. The speed of completion is constant from all the producers. So, the time of completion depends solely on the budget: the higher is the investment on a movie, the longer a producer takes to complete it.

The scheduling of movies is the main activity of producers and the focus of the model. At each time step, producers compute an index of competition for the following weeks. This index is the difference between the normalized expected number of movies on release and the normalized budget of the top ten films on release (ranked by budget). Then, producers decide which kind of competition index fit the movie. This choice involves risk sensitivity of the producers and features of the movie (in terms of quality and budget). Thirdly, producers pick the moment in the following two years with the minimum absolute difference between the desired competition index and the expected competition index and schedule the movie for that date.

The last feasible action for a producer is the retiring of a movie. It happens only when a combination of factors are present:

1. the audience in a week is less than the audience in the week before.
2. the movie is on release for at least three weeks
3. the potential market is below a certain share of the initial potential market.
4. the box office result of the movie is below the average, fixed for the budget (so that a small movie is not supposed to obtain the same result of a blockbuster).

The modelling of the audience does not consider heterogeneity in the preferences. It means that each individual of the audience is not interested in attending movies of a specific gender

but only "famous movies" (with high budget) and "good movies" (with high quality). For this reason, it is possible to define the potential initial market for each movie, starting from quality and budget. This value is a fraction of a total potential market, which stands for the overall number of spectators that would go to the cinema to see a movie. In this model, the total number of potential spectators is constant. The specific relationship between budget, quality and box office results is deepened in the calibration section. The audience of a movie also depends on the competition that it is facing. The competition affects the weekly box office of a film only if the sum of the potential markets of the movies on release that week is above the total number of potential spectators. In this case, the audience is distributed between all the possible movies using their residual potential spectators. Mathematically, it is

$$d_i = \frac{pa_i}{\sum pa}$$

with d_i attendance of a movie i at time t , pa_i potential audience of movie i at the time i , and $\sum pa$ sum of the potential audience of all the movies on release at the time t .

The model has two outputs: the normalized expected number of movies on release and the normalized box office results of the top ten movies on release (ranked by box office). The scheduling of the model is sequential. The order of the producers changes for each time step to guarantee realism and not advantage any specific producers in the scheduling decisions. Figure 2 depicts the scheduling process.

The simulation model was implemented using Python 3.8, and every simulation run on a Windows machine equipped with a 3.30GHz Intel(R) Core(TM) i5-4590 CPU and 4.0 GB RAM.

4. Calibration

This model represents a real-world system. Hence, to achieve valid results, it was necessary to initialize it with actual data. Besides, it was possible to calibrate its global behaviour to obtain a trend comparable to the one observed in the US movie market. These two phases differed in relevance, methodology and were sequential. For these reasons, each subsection of this paragraph outlines a specific calibration step.

4.1. Initial Calibration

This phase consisted of reproducing authentic producers in the model and adjusting their behaviour according to real-world data. It drew on a database of 4011 movies collected from IMDb. The database included titles, box office results, budgets, and production houses. Lately, we estimated the attendance for each movie from the box office results and the MPA THEME 2019 report [4]. Using this database, we were able to perform the following activities:

1. identification and selection of the real producers working on the US movie market in the time range between 2000 and 2019.
2. identification of the budget invested by each producer for every movie;
3. identification of the distribution of budgets for each selected producer;

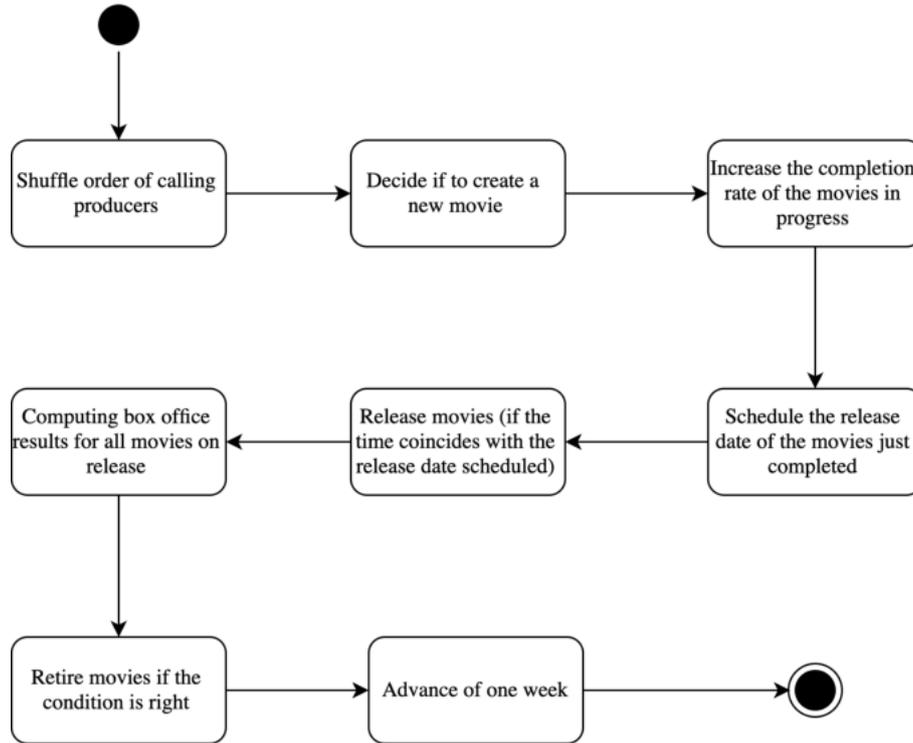


Figure 2: Scheduling process of the model for each time step

4. identification of the box office result for every movie released by selected producers;
5. identification of a relationship between the budget of a movie and the box office result;
6. identification of a relationship between budget and time of production.

Activities 2 and 4 were automatic, while activity 6 followed a pre-existent studio [32]. The rest of the paragraph deepens the other activities.

4.1.1. Producers identification and selection

The database contained 263 producers. We were able to characterize each of them for the number of movies released in the time range between 2000 and 2019. Then, we selected only the 27 producers with at least 20 releases (so, with an average of movies per year equal to or greater than 1).

4.1.2. Budget distribution for producers

A preliminary analysis of the probability distribution of the movies suggests that budgets are statistically distributed in several ways, and none of them was normal. Therefore, it was necessary to identify a set of possible probability distributions and test how good they would fit for the movies released by each producer. We tested gamma, power-law and uniform distribution. Table 2 shows the results of the analysis.

Table 2

Distribution of probability distribution of budgets for producers, calibrated on real data

Distribution	Occurrences
Gamma	19
Power law	6
Uniform	2

For each producer, we selected the probability distribution that fitted the best the data from the database.

4.1.3. Relationship between budget and box office results

Previous studies suggested that there was at least a correlation between the movies budgets and box office results [1, 2]. Using our dataset, we aimed to explain this relationship during the specific time range we examined. Moreover, we intended to address which share of the success derived from the budget. We estimated the other part was descending from other factors (such as genre, the RPAA code, the cast, the director and the overall quality of the product). To obtain this information, we developed a linear regression between the box office and the attendance of each movie in the database. We found out a linear relationship that explains the 54.1% of the variability of the result. We considered the rest of the variability to be a consequence of other factors, generically modelled as "movie quality" in this work.

4.2. Behaviour Calibration

In the introduction, we stated that this work aimed at reproducing a counterphased global behaviour observable on the aggregate time-series of the US movie market between the total number of movies on release and the box office results of the top 10 ranking movies. After the initial calibration, it was already possible to simulate the model imposing to producers random risk preferences. We noticed that only in some simulations the global behaviour of the agent-based model was comparable with the observed in the real world. Consequently, we decided to calibrate the individual behaviour of the producers to minimize the difference between the real-time series and the ones resulting from the simulation. Especially, we picked the time series shown in Figure 1. We called r the normalized number of movies on release in a given week (blue line) and as b the normalized values of the sum of the box office results for the top 10 movies in a certain week, ordered per box office result (red line). The calibration aims at replicating the following features:

1. the number of periods in which $r > b$.
2. the mean difference between b and r when $b > r$.
3. the number of intersections between b and r .
4. the mean value of r
5. the mean value of b

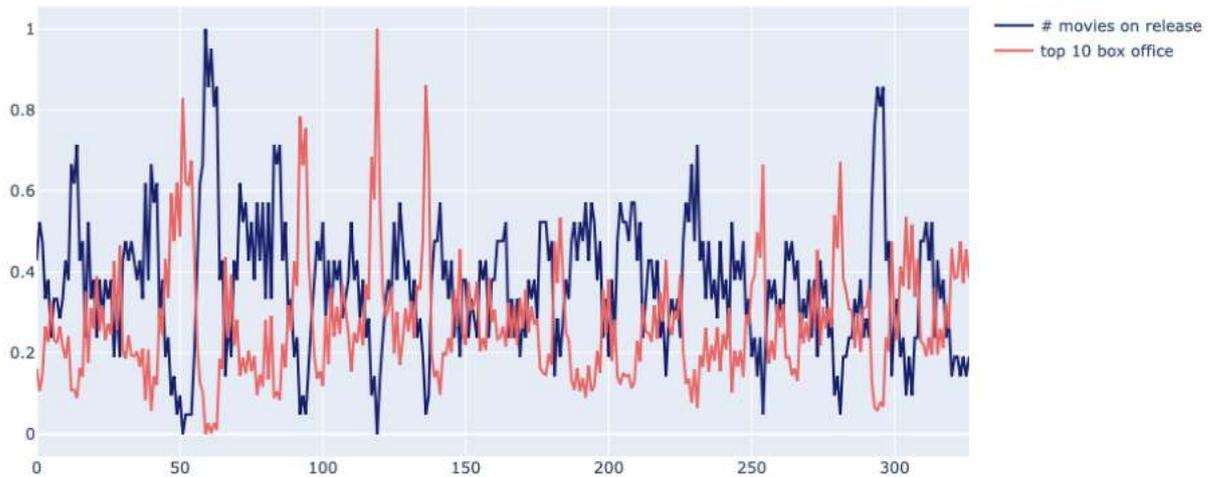


Figure 3: Simulated data time series of the number of movies on release each week and of the box office results of the top 10 ranking movies for box office (both values are normalized)

We employed a genetic algorithm to complete this activity. The utility function minimized was the distance between the five features shown above between the simulated data and the US movie market data. The algorithm ran for 200 generations with a population of 500 individuals. The algorithm simulated each member of the population 20 times.

5. Results

The simulation and the calibration of the model allowed us to achieve two main results:

1. the replication of the macro behaviour
2. specific risk sensitivity assigned to each producer, and the relationship between that and its features

5.1. Behaviour Replication

The macro behaviour replicated can be observed in Figure 3.

Figure 3 presents the same feature as Figure 1 in terms of counterphased behaviour and phase alternation. Therefore, we found that the macro behaviour present in the US movie market could be (at least partially) replicated by simply imposing basic rules regarding the presence of risk-sensible scheduling. This result highlighted the importance of risk preferences in the decision process of entities, especially when dealing with complex environments and trade-offs. While this result is domain-specific, it is comparable to pre-existent results in the literature related to different application domains [33, 6, 5]. Nevertheless, there were some differences between the two curves. It is observable that the periodicity of the seasonality and the amplitude of some sections varied. We proposed that the origin of these differences could be the cognitive simplicity of the producers, which, between the other limitations, do not have memory. For example, producer agents did not remember which moment of the year was best suited for

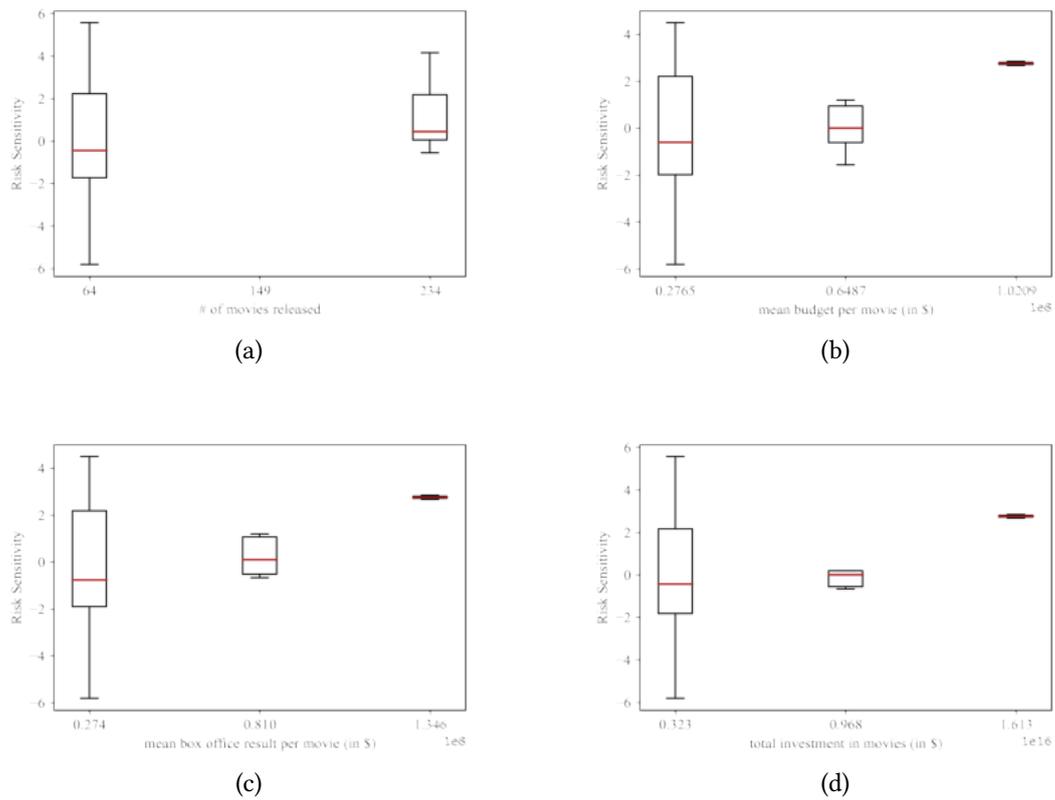


Figure 4: Distribution of risk sensitivity of producers per (a) total number of movies released, (b) mean budget of movies, (c) mean box office result of movies and (d) total investment in movies

releasing high budget movies. Therefore, the seasonality could not precisely fit real data. The difference in magnitude also descended from cognition simplicity. Producers did not know the plans of the other producers before a competitor movie became scheduled. It affected the regularity of the height of the spikes for both r and s .

5.2. Risk sensitivity of producers

This conclusive analysis compared the risk sensitivities of producers resulting from the behaviour calibration with the US movie market data about producers' behaviour gathered during the individual calibration. Figure 4 exhibits the outcomes of this study. In each subfigure, the 27 production studios are represented in one of the three boxplots. Each box included the point located in one-third of the total area.

Each subfigure deemed a different variable: the total number of movies released, the mean budget per movie, the mean box office result per movie, and the total amount of investment in dollars, which is the sum of the budgets of all the movies released. We computed every variable for each producer for the time range 2000-2019.

The figures owned two notable features. First, the mean risk sensitivity increased with the growth of each variable. It was interesting for a twofold reason. On the one side, it was coherent with previous findings in risk preferences literature, for which entities tend to be more cautious when the stack increases [34, 5]. On the other, it was consistent with empirical observations related to the kind of movies released. In the last ten years, a substantial number of released were remakes or sequels of previous works [35]. It was a behaviour adopted mainly by big production studios, which tries to minimize the risk of a flop. Hence, our analysis is coherent with these findings. The second remarkable characteristic regards the variability of the results. With higher stakes, the variability of the risk sensitivity decreased. We suggested that it could be a consequence of the market structure, for which the specific macro behaviour could arise when the small producers have high variability in risk preference while big producers are similarly risk-averse.

6. Conclusions

This paper provides a possible explanation to the seasonal and counterphased behaviour of the time series of the number of movies on release and the sum of the box office results for the top 10 movies released (ranked for box office result) in the US movie market data. In this work, we replicate it by developing and calibrating an agent-based model of the US cinema market. Moreover, we identify relationships between the calibrated risk sensitivity and the real features of the producers, for which we suggest that:

1. the higher is the amount of budget invested by a producer, the higher is its risk aversion.
2. the higher are the investments, the lower is the expected distance of its risk sensitivity from the mean risk sensitivity of other producers in the same investment range.

The work has some limitations and some related future developments:

- the purpose of the model was to understand the reason for the counterphased seasonality of the number of released and the box office results of the top 10 movies (ranked for box office success in a certain week). The model proposes a potential interpretation of this phenomenon, but it did not show the adaptation of the preferences. This specific issue could be addressed by the following work.
- The movie market is calibrated not with all the possible movies but only with the principal. While we consider it sufficient to study the dynamics of the most important producers, there is the possibility of an underestimation of the impact of smaller producers on the overall behaviour.
- we considered the US movie market between 2000 and 2019. So, the insurgence of the COVID-19 pandemic is not included. Future studies could analyze the effect of a disruptive event that lead to a 32 billion dollars loss globally [36].
- the behaviour of producers were modelled under the assumption that they maximize the profit. Nevertheless, we do not analyze the economic performance of producers. Possible future developments include the analysis of each risk strategy on the producers' performance. What is more, it could be investigated if and how the distribution of risk

preferences in the producers on the competitive landscape affects the overall profitability of the whole market.

References

- [1] S. M. Brewer, J. M. Kelley, J. J. Jozefowicz, A blueprint for success in the US film industry, *Applied Economics* (2009). doi:10.1080/00036840601007351.
- [2] R. A. Nelson, R. Glotfelty, Movie stars and box office revenues: An empirical analysis, *Journal of Cultural Economics* (2012). doi:10.1007/s10824-012-9159-5.
- [3] A. De Vany, W. D. Walls, Uncertainty in the movie industry: Does star power reduce the terror of the box office?, *Journal of Cultural Economics* (1999). doi:10.1023/A:1007608125988.
- [4] 2019 Theme Report, Technical Report, Movie Producer Association, 2020.
- [5] A. Hintze, R. S. Olson, C. Adami, R. Hertwig, Risk sensitivity as an evolutionary adaptation, *Scientific Reports* (2015). doi:10.1038/srep08242.
- [6] G. G. Szpiro, The emergence of risk aversion, *Complexity* (1997). doi:10.1002/(SICI)1099-0526(199703/04)2:4<31::AID-CPLX8>3.0.CO;2-3.
- [7] C. Prylvck, Front office box office, and artistic freedom: An aspect of the film industry 1945-1969, *Journal of Popular Film* (1974). doi:10.1080/00472719.1974.10661743.
- [8] S. Basuroy, S. Chatterjee, S. Abraham Ravid, How Critical are Critical Reviews? The Box Office Effects of Film Critics, Star Power, and Budgets, *Journal of Marketing* (2003). doi:10.1509/jmkg.67.4.103.18692.
- [9] Y. Liu, Word of mouth for movies: Its dynamics and impact on box office revenue, 2006. doi:10.1509/jmkg.70.3.74.
- [10] T. Liu, X. Ding, Y. Chen, H. Chen, M. Guo, Predicting movie Box-office revenues by exploiting large-scale social media content, *Multimedia Tools and Applications* (2016). doi:10.1007/s11042-014-2270-1.
- [11] M. Hur, P. Kang, S. Cho, Box-office forecasting based on sentiments of movie reviews and Independent subspace method, *Information Sciences* (2016). doi:10.1016/j.ins.2016.08.027.
- [12] J. H. Lee, S. H. Jung, J. H. Park, The role of entropy of review text sentiments on online WOM and movie box office sales, *Electronic Commerce Research and Applications* (2017). doi:10.1016/j.eierap.2017.03.001.
- [13] L. Zhang, J. Luo, S. Yang, Forecasting box office revenue of movies with BP neural network, *Expert Systems with Applications* (2009). doi:10.1016/j.eswa.2008.07.064.
- [14] R. Parimi, D. Caragea, Pre-release box-office success prediction for motion pictures, in: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013. doi:10.1007/978-3-642-39712-7_44.
- [15] L. N. Yasnitsky, I. A. Mitrofanov, M. V. Immis, Intelligent System for Prediction Box Office of the Film, in: *Lecture Notes in Networks and Systems*, 2020. doi:10.1007/978-3-030-22493-6_3.
- [16] M. Mestyán, T. Yasseri, J. Kertész, Early Prediction of Movie Box Office Success Based on

- Wikipedia Activity Big Data, PLoS ONE (2013). doi:10.1371/journal.pone.0071226. arXiv:1211.0970.
- [17] J. Ribera, S. Sieber, How uncertainty and risk management impacts the success of spanish film projects, *Journal of Media Business Studies* (2009). doi:10.1080/16522354.2009.11073495.
- [18] M. B. Von Rimscha, Managing risk in motion picture project development, *Journal of Media Business Studies* (2009). doi:10.1080/16522354.2009.11073496.
- [19] G. Bi, D. E. Giles, Modelling the financial risk associated with U.S. movie box office earnings, *Mathematics and Computers in Simulation* (2009). doi:10.1016/j.matcom.2008.04.014.
- [20] K. Satoh, S. Matsubara, Box-Office Prediction Based on Essential Features Extracted from Agent-Based Modeling, in: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021. doi:10.1007/978-3-030-69322-0_31.
- [21] R. Kumar Pan, S. Sinha, The statistical laws of popularity: Universal properties of the box-office dynamics of motion pictures, *New Journal of Physics* (2010). doi:10.1088/1367-2630/12/11/115004. arXiv:1010.2634.
- [22] F. Gutierrez-Navratil, V. Fernandez-Blanco, L. Orea, J. Prieto-Rodriguez, How do your rivals' releasing dates affect your box office?, *Journal of Cultural Economics* (2014). doi:10.1007/s10824-012-9188-0.
- [23] F. Gutiérrez-Navratil, V. Fernández-Blanco, L. Orea-Sánchez, J. Prieto-Rodríguez, Do movie majors really collude? Indirect evidence from release schedules, *Hacienda Publica Espanola* (2017). doi:10.7866/HPE-RPE.17.2.1.
- [24] S. A. Delre, T. L. Broekhuizen, W. Jager, The effect of social influence on market inequalities in the motion picture industry, in: *Proceedings of the 4th Conference of the European Social Simulation Association, ESSA 2007*, 2007.
- [25] S. A. Delre, C. Panico, B. Wierenga, Competitive strategies in the motion picture industry: An ABM to study investment decisions, *International Journal of Research in Marketing* (2017). doi:10.1016/j.ijresmar.2016.05.005.
- [26] A. De Vany, C. Lee, Quality signals in information cascades and the dynamics of the distribution of motion picture box office revenues, *Journal of Economic Dynamics and Control* (2001). doi:10.1016/S0165-1889(00)00037-3.
- [27] C. Iasiello, Underrepresentation of minorities in hollywood films: An agent based modeling approach to explanations, in: *Proceedings - Winter Simulation Conference, 2017*. doi:10.1109/WSC.2017.8248215.
- [28] T. L. Broekhuizen, S. A. Delre, A. Torres, Simulating the cinema market: How cross-cultural differences in social influence explain box office distributions, *Journal of Product Innovation Management* (2011). doi:10.1111/j.1540-5885.2011.00792.x.
- [29] D. C. Chisholm, V. Fernández-Blanco, S. Abraham Ravid, W. David Walls, Economics of motion pictures: the state of the art, 2015. doi:10.1007/s10824-014-9234-1.
- [30] L. Cabral, G. Natividad, Box-Office Demand: The Importance of Being #1, *Journal of Industrial Economics* (2016). doi:10.1111/joie.12095.
- [31] E. Bonabeau, Agent-based modeling: Methods and techniques for simulating human systems, *Proceedings of the National Academy of Sciences of the United States of America*

- (2002). doi:10.1073/pnas.082080899.
- [32] stephenfollows.com, How long does the average hollywood movie take to make?, 2018. URL: <https://stephenfollows.com/how-long-the-average-hollywood-movie-take-to-make/>.
 - [33] R. Zhang, T. J. Brennan, A. W. Lo, The origin of risk aversion, *Proceedings of the National Academy of Sciences of the United States of America* (2014). doi:10.1073/pnas.1406755111.
 - [34] S. Mishra, Decision-Making Under Risk: Integrating Perspectives From Biology, Economics, and Psychology, *Personality and Social Psychology Review* (2014). doi:10.1177/1088868314530517.
 - [35] B. Bohnenkamp, A. K. Knapp, T. Hennig-Thurau, R. Schauerte, When does it make sense to do it again? An empirical investigation of contingency factors of movie remakes, *Journal of Cultural Economics* (2015). doi:10.1007/s10824-014-9221-6.
 - [36] D. Hancock, M. Signorelli, S. Henschel, Viewpoint report: Movie windows – adapting for the future, 2020. URL: <https://omdia.tech.informa.com/OM013236/Viewpoint-Report-Movie-Windows--Adapting-for-the-Future>.

A Curriculum–Based Reinforcement Learning Approach to Pedestrian Simulation

Thomas Albericci¹, Thomas Cecconello¹, Alberto Gibertini¹ and Giuseppe Vizzari¹

¹*Department of Informatics, Systems and Communication (DISCo), University of Milano-Bicocca, Milan, Italy*

Abstract

Reinforcement Learning represents a way to train an agent situated in an environment what to do to maximise an accumulated numerical reward signal (received by the environment as a feedback to every chosen action). Within this paper we explore the possibility to apply this approach to pedestrian modelling: pedestrians generally do not exhibit an optimal behaviour, therefore we carefully defined a reward function (combining contributions related to proxemics, goal orientation, basic wayfinding considerations), but also a particular training *curriculum*, a set of scenarios of growing difficulty supporting the incremental acquisition of proper orientation, walking, and pedestrian interaction competences. The paper will describe the fundamental elements of the approach, its implementation within a software framework employing Unity and ML-Agents, describing the promising achieved simulation results.

Keywords

agent-based simulation, pedestrian simulation, reinforcement learning, curriculum learning

1. Introduction

Reinforcement Learning (RL) [1] is a machine learning approach that is being growingly investigated as way to achieve autonomous agents, where the acceptance of the term “autonomous” is closer to Russell and Norvig’s [2] than the most widely adopted ones in agent computing. Russell and Norvig state that:

A system is autonomous to the extent that its behavior is determined by its own experience

RL represents a way to train an agent situated in an environment what to do to maximise an accumulated numerical reward signal (received by the environment as a feedback to every chosen action). The agent is provided with a model of perception and action, but besides these modelling elements and the reward function, the approach can autonomously explore the space of potential agent behaviours and converge to a policy (i.e. a function mapping the state and perception to an appropriate action to be carried out in that context).

A certain amount of initial knowledge (in an analogy to built-in reflexes in animals and humans, but also internalized norms, rules, and even ways to evaluate the degree of acceptability

WOA 2021: Workshop “From Objects to Agents”, September 1–3, 2021, Bologna, Italy

✉ t.albericci1@campus.unimib.it (T. Albericci); thomas.cecconello@unimib.it (T. Cecconello);

a.gibertini@campus.unimib.it (A. Gibertini); giuseppe.vizzari@unimib.it (G. Vizzari)

ORCID 0000-0002-7916-6438 (G. Vizzari)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

of a state of affairs) is therefore considered by the approach, but it should be sided by the ability to learn, to adjust one's behaviour to achieve a better performance. RL approaches, reinvigorated by the energy, efforts, and promises brought by the *deep learning* revolution, seems one of the most promising ways to investigate how to provide an agent this kind of autonomy. On a more pragmatic level, recent developments and results in the RL area suggest that this approach might even be a promising alternative to current agent-based approaches to the modeling of complex systems [3]: whereas currently behavioral models for agents are carefully hand crafted, often following a complicated interdisciplinary effort involving different roles and types of knowledge, as well as validation processes based on the acquisition and analysis of data describing the studied phenomenon, RL could simplify this work, focusing on the definition of an environment representation, the definition of a model for agent perception and action, and defining a reward function. The learning process could, in theory, be able to explore the potential space of the *policies* (i.e. agent behavioral specifications) and converge to the desired decision making model. While the definition of a model of the environment, as well as agent perception and action, and the definition of a reward function are tasks requiring substantial knowledge about the studied domain and phenomenon, the learning process could significantly simplify modeler's work, and at the same time it could solve issues related to model calibration. Although some relevant related work can be found in the literature (in particular [4]), results achieved so far highlight significant limitations, especially in the capability of generalization of the training phase: this is a very important aspect for this kind of application, not just because it is inconvenient to pay this computational cost for every scenario to be analyzed, but also due to the fact that results could not be actually comparable, since they would be achieved with different simulation models.

Within this line of work, and building on preliminary efforts [5], this paper describes an experimentation of this approach to pedestrian modelling. Whereas is RL agents learn how to behave to optimize their expected cumulative reward, pedestrians generally do not exhibit an optimal behaviour. Therefore we carefully defined a reward function (combining contributions related to proxemics, goal orientation, but also basic wayfinding considerations). We also employed a particular training *curriculum* [6], a set of scenarios of growing difficulty supporting the incremental acquisition of proper orientation, walking, and pedestrian interaction competences.

The paper will describe the fundamental elements of the approach, its implementation within a software framework employing Unity¹ and ML-Agents², describing the promising achieved simulation results: in particular, we will show that the proposed approach is able to produce plausible results in environments that were not used for sake of training, so the approach seems promising at least in terms of generality. We will finally discuss the current limits of the approach, and our current implementation, as well as ongoing future developments.

¹<https://unity.com>

²<https://github.com/Unity-Technologies/ml-agents>

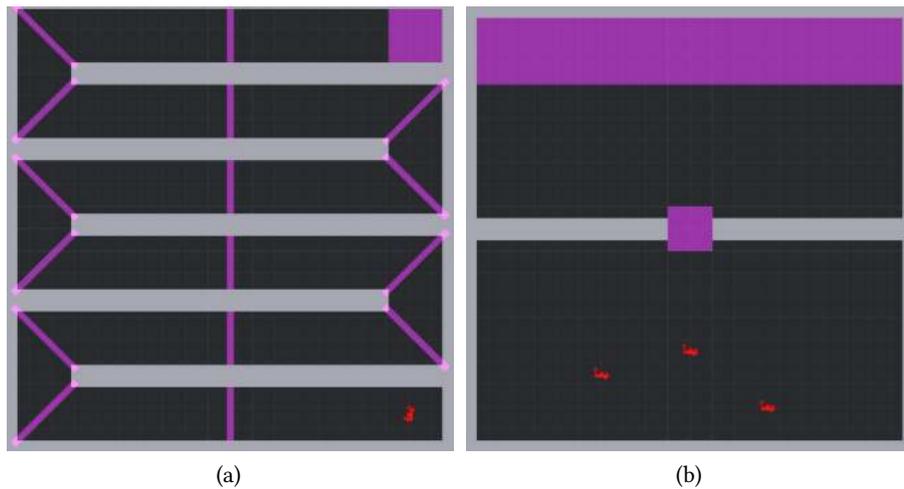


Figure 1: (a) 'Turns' environment and annotations, and (b) 'Unidirectional Door' environment and annotations.

2. The Model

2.1. Representation of the Environment

For sake of simplicity in this experimental study environments are bound to be squares of 20×20 metres surrounded by walls. The smaller squares (of 1×1 metre) that can be seen in the figures presented later on are just for sake of allowing a simpler appraisal of distances. Gray objects are walls, obstacles and anything that agents perceive as 'Wall'.

Violet rectangles are intermediate and final goals. These markers (in the vein of [7]), do not hinder the possibility of moving through them, and they are essentially a modeling tool to support agent's navigation in the environment. In fact, one of the goals of the work is to provide an alternative to Unity's path finding and (more generally) pedestrian agent control mechanisms. Later we will describe agents' perceptive model, but we anticipate that they are able to perceive these markers and to select intermediate or final movement targets; we will also see that reaching intermediate or final targets will also influence agent's reward.

Environments must therefore undergo a preparation phase before being actually used in the proposed approach; an example of an environment annotated with this rationale is shown in Figure 1(a). In this case, the targets in the middle of the horizontal corridors create an affordance for agents to move towards that direction although the actual bend, at the end of the corridor, is fairly distant, and this could confuse agents during the training phase. Moreover, oblique targets in the bends guide agents in the change of direction, also helping them to achieve a plausible trajectory [8]. Figure 1(b) shows instead an environment in which a door (an open one, of course) is present: in this case, the target is used to guide agents passing through the opening, since the final target is obstructed and not perceivable from a large portion of the Southern room.

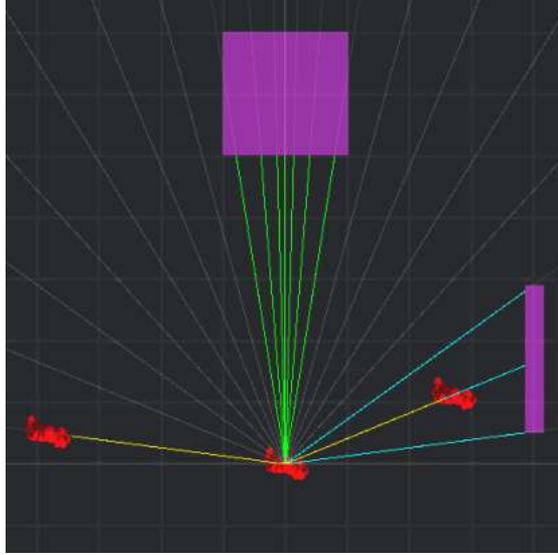


Figure 2: Rays and provided information: yellow = agent, cyan = intermediate target, green = final target, transparent = wall or none of the others. Pedestrian agents are depicted in red.

2.2. Agent Perception

Agents are provided with a limited set of *projectors* of rays, each extending up to a certain distance (10 m in these experiments) and providing specific information about what is “hit” by the ray and the associated distance from the agent.

Projectors (and therefore rays) are not uniformly distributed around the agent, but they are more densely present in front of the pedestrian, to loosely resemble real human visual perception.

The angle between the rays and the facing direction of an agent (both positive and negative) follows the rule described in Equation 1:

$$\alpha_i = \text{Min}(\alpha_{i-1} + \delta * i, \text{max_vision}) \quad (1)$$

where δ has been set to 1.5, max_vision to 90 and α_0 to 0. As a consequence, projectors emit rays at $0^\circ, \pm 1.5^\circ, \pm 4.5^\circ, \pm 9^\circ, \pm 15^\circ, \pm 22.5^\circ, \pm 31.5^\circ, \pm 42^\circ, \pm 54^\circ, \pm 67.5^\circ, \pm 82.5^\circ$ and $\pm 90^\circ$. Figure 2 graphically depicts this distribution.

The overall number of projectors and rays would therefore be 23, but since the information associated to and conveyed by rays is different for different objects we actually have several projectors for each angle, and therefore each agent actually has 46 rays.

The overall agent’s observation is summarized in Table 1. To improve the performance of neural networks typically employed in recent RL algorithms all observations have been normalized in the interval $[0,1]$. In particular, for normalization of walking speed we consider the maximum velocity for walking agents to be 1.7 m/s.

Information about Walls and Targets is provided to support basic wayfinding, whereas information about Agents and Walls is more detailed (including also the walking direction and

Table 1
summary of agent’s observations.

Type of observation	Observation	Value
Intrinsic	Own speed	Number
Walls and Targets	Distance	Number
	Type/Tag	One Hot Encoding
Agents and Walls	Distance	Number
	Type/Tag	Boolean
	Direction	Number
	Speed	Number

speed, in case the ray ‘hits’ an agent) and it is provided to support more fine grained collision avoidance.

2.3. Action space

Each agent is provided with an individual desired velocity that is drawn from a normal distribution with average of 1.5 m/s and a standard deviation of 0.2 m/s. Each decision, and for these experiments we decided to grant agents three decisions per second (in line with [9], combining cognitive plausibility, quality of the achieved results, and computational costs), determines a potential change in its velocity and this is basically what agent’s decision is all about for this model.

Agent’s action space has been therefore modeled as the choice of two (conceptually) continuous values in the [-1,1] interval that are used to determine a change in velocity vector, respectively for magnitude and direction. The first element, a_0 , causes a change in the walking speed defined by Equation 2:

$$speed_t = Max \left(speed_{min}, Min \left(speed_{t-1} + \frac{speed_{max} * a_0}{2}, speed_{max} \right) \right) \quad (2)$$

Where $speed_{min}$ is set to 0 and $speed_{max}$ is set to 1.7 m/s. According to this equation the agent is able to reach a complete stop or the maximum velocity is two actions (i.e. about 0.66 s).

The second element of the decision, a_1 , determines a change in agent’s direction according to Equation 3:

$$\alpha_t = \alpha_{t-1} + a_1 * 20 \quad (3)$$

The walking direction can therefore change 20°each 0.33 s, that is plausible for normal pedestrian walking, but would be probably not reasonable for modeling running and/or sport related movements.

2.4. Reward Function

The reward function is a crucial element for a RL approach: it represents the feedback signal guiding the learning process, in a certain sense it represents a (weaker) substitute for labels in supervised learning. Moreover, here we deal with a complex form of decision making, with conflicting tendencies that are generally reconciled quickly, almost unconsciously, in a reasonable/explainable way (in retrospective) by the typical pedestrian, in a combination of individual and collective intelligence, that however leads to sub-optimal overall performance (see, for instance, the above cited [7] but also [10]).

Given the above considerations, we hand-crafted a reward function, initially in terms of components, i.e. factors generally influencing pedestrian behaviour. Later on we performed a sort of initial tuning of the related weights defining the relative importance of the different factors. A sensitivity analysis was not performed and it would be object of future works.

The overall reward function is defined in Equation 4:

$$\text{Reward} : \left\{ \begin{array}{ll} +6 & \text{Final target reached} \\ +0.5 & \text{Intermediate target reached} \\ -1 & \text{Reached a previously reached intermediate target} \\ -0.5 & \text{No target in sights} \\ -0.5 & \text{Agent in very close proximity - } < 0.6 \text{ m} \\ -0.005 & \text{Agent in close proximity } < 1 \text{ m} \\ -0.001 & \text{Agent in proximity } < 1.4 \text{ m} \\ -0.5 & \text{Wall in proximity } < 0.6 \text{ m} \\ -0.0001 & \text{Each step done} \\ -6 & \text{Reached the end of steps per episode} \end{array} \right. \quad (4)$$

The only ways to increase the cumulative reward are therefore the reaching of intermediate or final targets. However, reaching targets that have been previously visited brings a negative reward, since it would imply moving back from the final goal, and it makes it much less reasonable to try to “exploit” the reward to reach a formally reasonable but totally implausible policy (i.e. reach as many intermediate targets before reaching the final one before the end of the episode). Negative rewards thus are used to suggest that some actions should not be chosen unless they eventually lead to the final goal (and unless better alternatives do the same): a small negative reward granted due to the simple passage of time is usual, it pushes agents to avoid standing still and to actively look for solutions, but we also have negative rewards due to proxemics [11], and to penalize walking too close to walls (again, unless necessary). Finally, the penalization to actions leading to a position from which no target (either intermediate or final) can be seen stimulates agents to pursue the goals; one could wonder if having instead a small bonus for actually seeing a target would work analogously: all positive rewards should however be taken very carefully, since they can lead to pathological behaviours. In this case, in very complex scenarios, an agent might learn to find a target and stand still, achieving a relatively small bonus for each decision of the episode.

2.5. Adopted RL algorithm

For this research and experimentation we adopted Proximal Policy Optimization (PPO) [12], a state-of-the-art RL policy-based algorithm provided by ML-Agents. PPO is a policy gradient algorithm which works by learning the policy function π directly. These methods have a better convergence properties compared to dynamic programming methods, but need a more abundant set of training samples. Policy gradients work by learning the policy's parameters through a policy score function, $J(\Theta)$, through which is possible to apply gradient ascent to maximize the score of the policy with respect to the policy's parameters, Θ . A common way to define the policy score function is through a loss function:

$$L^{PG}(\Theta) = E_t[\log\pi_{\Theta}(a_t|s_t)]A_t \quad (5)$$

which is the expected value of the log probability of taking action a_t at state s_t times the advantage function A_t , representing an estimate of the relative value of the taken action. As such, when the advantage estimate is positive, the gradient will be positive as well; through gradient ascent the probability of taking the correct action will increase, while decreasing the probabilities of the actions associated to negative advantage, in the other case.

The goal of the work was essentially to evaluate the adequacy of the approach to the problem of achieving a proper pedestrian simulation model and we did not yet analyze the performance of different RL algorithms, something that is object of future works.

3. Curriculum Learning

3.1. Rationale of the Approach

Curriculum Learning, introduced in [6], represents a strategy within machine learning initially devised with the aim of reducing the training times. The rationale is to present examples in a specific order of increasing difficulty during training, illustrating gradually more concepts and more complications to the decision. Later on, it has been employed more specifically as a *transfer learning* technique in RL and Multi-Agent RL [13]: the agent can exploit experiences acquired carrying out simpler tasks while training to solve more complex ones, in an *intra-agent* transfer learning scheme. In some situation it was also reported to support a better generalization of the overall training process [14]: achieving a good level of generalization of the acquired experience was also extremely important for our problem, since pedestrian simulation generally implies analysing the implications of different, alternative designs on the same crowding condition, without having to perform training for every specific design (which would lead to achieve incomparable results, since they would be achieved by means of different pedestrian models).

A naive application of a curriculum approach, however, initially led to issues somewhat resembling the so-called *vanishing gradient* problem [15]: technically here we do not have a recurrent neural network (or an extremely deep one like those employed for classification of images trained on huge annotated datasets) but, as we will show later on, the training is relatively long and the "oldest experiences" would be overridden by the more recent ones. The finally adopted approach, therefore, proceeds training agents in a set of scenarios of growing

complexity, one at a time, but it also provides a final retraining in a selected number of earlier scenarios before the end of the overall training, to refresh previously acquired competences.

3.2. Details of the Curriculum

Starting from the above considerations, we defined a specific curriculum for RL-pedestrian agents based on this sequence of tasks of increasing complexity that are sub-goals of the overall training:

- Steer and walk towards a target;
- Steer to face target;
- Reach the target in narrow corridors;
- Walk through bends avoiding walking too close to walls;
- Avoid collisions with agents walking in the same direction;
- Avoid collisions with agents walking in conflicting directions;
- Combine all behaviours.

We defined this sequence thanks to expertise in the context of pedestrian simulation, as well as to a preliminary experimental phase (for instance the second step – steering to face a target – was introduced quite late, when we realized that, as a consequence of training in more geometrically complex scenarios, agents had sometimes difficulties in finding their targets when the environment was not essentially “guiding them”). It would be interesting to evaluate to which extent this sequence is robust, if it can be improved or if it is close to the optimum, but such an analysis was not performed at this stage of the research (we were interested in evaluating the adequacy of the approach and the possibility to achieve promising results on the domain of pedestrian simulation), and it is object of future works.

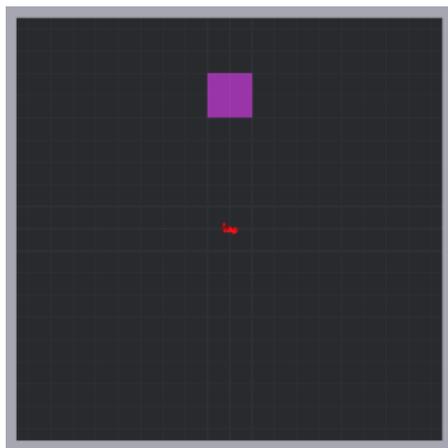
For sake of automation of the curriculum execution, we consider a step of the curriculum to be successfully completed whenever (i) a sufficiently high number of agents has been trained in the scenario and (ii) the average cumulative reward for trained agents, excluding the top and bottom 10% (for avoiding being excessively influenced by a small number outliers), exceeds a given threshold, specifically configured for every step of the curriculum.

We also included specific test scenarios, that is, environments that are not included in the training curriculum but that are used to evaluate the ability of agents to exhibit plausible behaviours in scenarios that were not experienced in the training phase, rather than just showing that they memorized the environments they had seen.

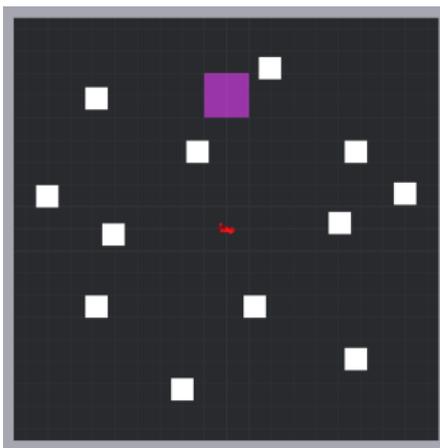
3.3. Training Environments

Table 2 reports the different environments that were defined for each of the sub-goals of the overall training. It also shows whose environment are included in the final retraining phase, that must be carried out before using the trained agents for simulation in new environments.

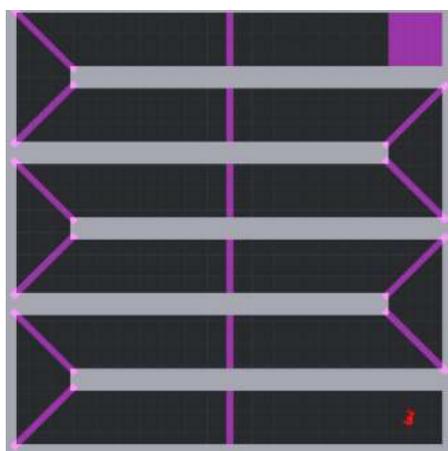
For sake of space, we cannot describe every environment and scenario included in the curriculum, but a selection of these training environments is shown in Figure 3. Several of these scenarios replicate experiments that were carried out with real pedestrians to study specific behaviours (see, e.g., [16] or [17]), although we currently did not investigate high-density



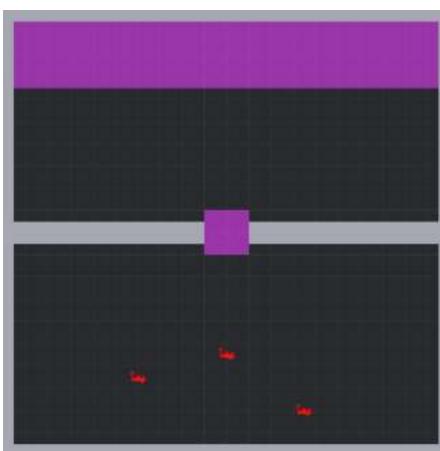
(a) Start Environment



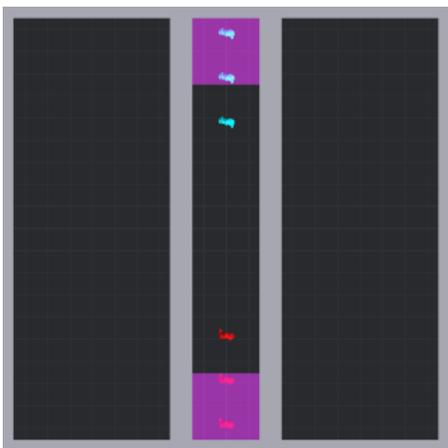
(b) Observe Environment



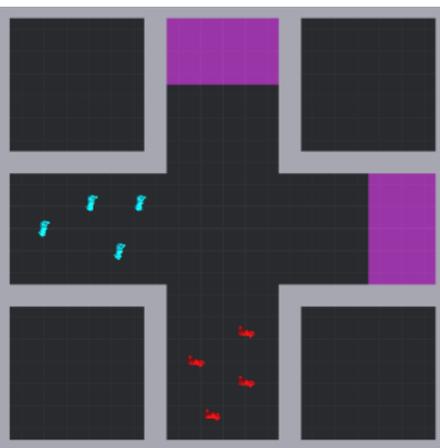
(c) Turns Environment



(d) Unidirectional Door Environment



(e) Corridor Environment



(f) Intersection Environment

Figure 3: A selection of training Environments.

Table 2
Training Environments Curriculum.

Behaviour	Environment	Retraining
Steer and walk towards a target	StartEz	×
	Start	✓
Steer to face target	Observe	✓
Reach the target in narrow corridors	Easy Corridor	×
Walk through bends avoiding walking too close to walls	Turns	×
	Turns with Obstacles	✓
Avoid collisions with agents walking in the same direction	Unidirectional Door	✓
Avoid collisions with agents walking in conflicting directions	Corridor	✓
	Intersection	✓
	T Junction	✓
Combine all behaviours	Crowded Bidirectional Door	✓

situations, that moreover seem difficult to simulate with a tool such as Unity (which includes 3D models for pedestrians and components for the management of physics that should be overridden for managing significant levels of density - e.g. higher than 1 pedestrian per square metre).

We also do not have the space for commenting the training in all of these scenarios, however we can highlight some stylized facts we did observe:

- within the Corridor Environment agents learn to walk in lanes that, due to the low density, are quite stable;
- the Turns and Turns with Obstacles Environments produce plausible results in terms of trajectories, but this is mostly due to the placement of intermediate target helping agents in having smooth and plausible paths (as suggested in subsection 2.1);
- all the environments in which agents had to face narrow passages were crucial in leading them to accept the trade off between choosing some actions leading to an immediate negative reward (i.e. passing close to a wall) and achieving a longer term positive reward (i.e. reaching the final target);
- all the environments in which agents had to interact with others were analogously crucial but for helping them understand how to properly balance the need of slowing down and sometimes even waiting (when steering is simply not possible or not sufficient) to avoid collisions, but still reach the final target.

```

behaviors:
Pedestrian:
  trainer_type: ppo
  hyperparameters:
    batch_size: 512
    buffer_size: 5120
    learning_rate: 0.003
    beta: 0.01
    learning_rate_schedule: constant
  network_settings:
    hidden_units: 256
    num_layers: 2
  reward_signals:
    extrinsic:
      gamma: 0.99
      strength: 1.0
  max_steps: 10000000000000000000
  time_horizon: 64

```

Listing 1: Training configuration file.

3.4. Training Configuration

Listing 1 reports the defined training configuration file³. The employed ML-Agents version we adopted is 0.25.1 for Python and 1.0.7 for Unity.

Once again, we were interested here in evaluating the adequacy of the approach, so we did not perform a systematic analysis of the effect of changing the different hyperparameters and this task will be object of future works. We just comment here some of the adopted choices:

- the neural network employed within the PPO algorithm is a fully connected network with 2 hidden layers of 256 nodes each; a bigger network leads to much longer training times but it does not improve the quality of the achieved results, whereas a smaller network does not converge to a reasonable policies;
- we employed a basic PPO without curiosity mechanisms [18], therefore we have essentially just extrinsic reward signals;
- we adopted a very high number for `max_steps` to let the curriculum guide the actual training, rather than predefined parameters. We also let `time_horizon` to the default value.

3.5. Reward Trend During Training

The preliminary tests we conducted before reaching this configuration for the curriculum, that were based on a single scenario or however that were based on curricula significantly

³Detailed descriptions of different fields are reported in https://github.com/Unity-Technologies/ml-agents/blob/release_16_docs/docs/Training-Configuration-File.md

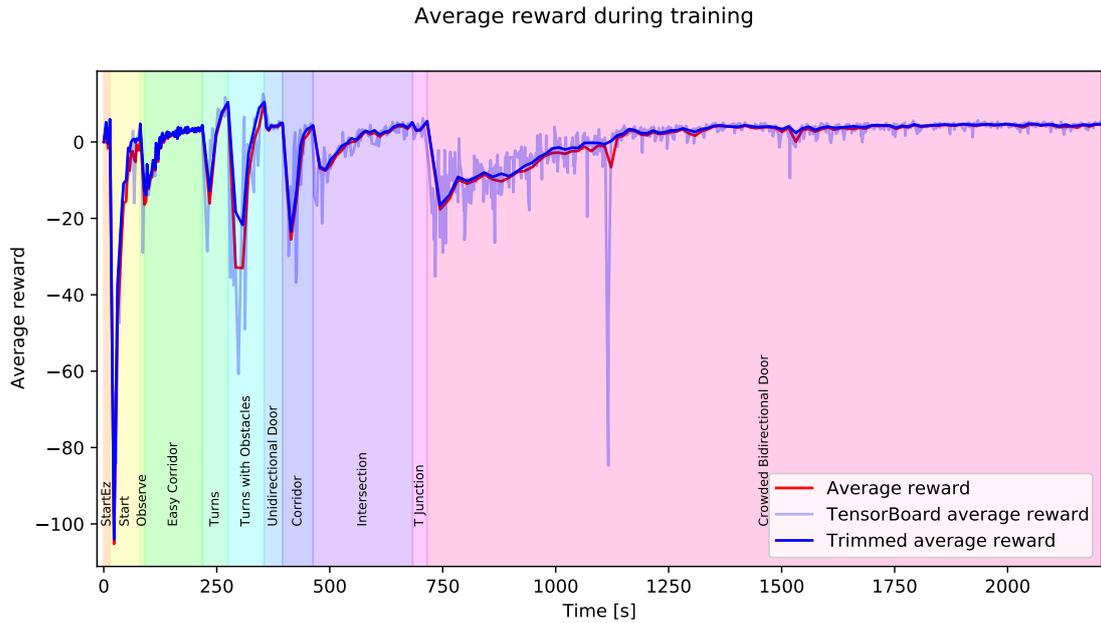


Figure 4: Average reward throughout training.

more compact than the one described above, were unsuccessful (at least they did not produce good results within the same time frame associated to training with this configuration for the curriculum).

The overall training time with the defined curriculum varies according to different factors, but on a Windows based PC employing an Intel Core i7-6820HL @ 2.70GHz, employing only the CPU⁴ would require around 37 minutes to reach the final retraining phase (which is significantly shorter). Technically, agents have been trained in 9 equal environments at the same time, with a Unity velocity set to 100 (i.e. one second of simulation execution corresponds to 100 simulated seconds). The available hardware would not allow further compression of simulated time, but future developments in the ML-Agents framework could bring significant improvements (especially if they would fully exploit GPUs), but we could need to change the training phase workflow.

Figure 4 shows the trend of the cumulative reward. The Tensorboard average reward is the raw measure provided by Tensorboard, while the *Average reward* is computed averaging out the cumulative reward achieved by agents in 36 episodes within an environment. The *Trimmed average reward* actually removes respectively the 10% top and 10% bottom performing episodes.

The different colors highlight the duration of the different scenarios of the curriculum: as expected the reward drops (sometimes dramatically) when agents change the environment, but through time the training converges. It also clearly shows that environments in which agents have more significant interactions are tougher for the training algorithm. We were

⁴The adopted version of ML-Agents suggests doing so, since it would not properly exploit a GPU.

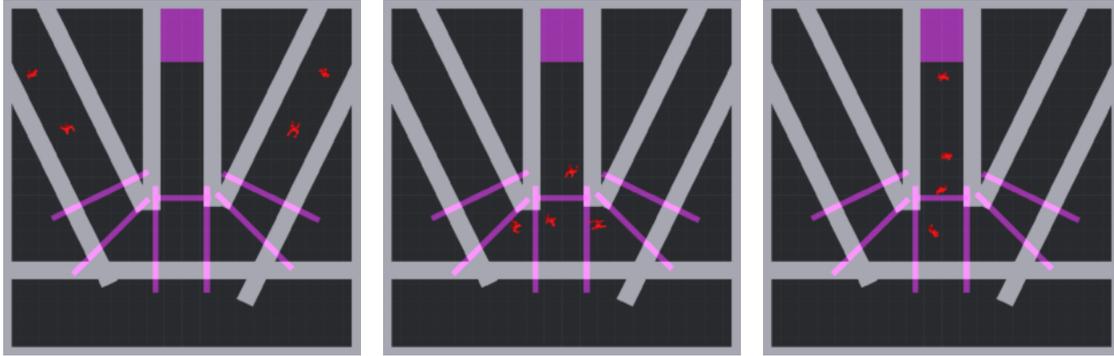


Figure 5: Anchor environment execution.

actually almost surprised by the fact that a vanilla PPO was able to successfully converge in such situations, that are much closer to situations that call for specific Multi-Agent RL algorithms. In these situations, the basic approaches often fail due the instability in the reward trend that depends on more factors outside the scope of control of the trained agent; specific reward functions that balance individual and aggregated level evaluation of the situation and new algorithms are typically employed. We also conducted an analogous experimentation considering groups of pedestrians, a situation that makes pedestrian to pedestrian interaction both more complex and much more frequent (essentially uniformly present in each step of the training) than the type described in the present work, and PPO was not able to converge. The description of this additional experimentation is out of the scope of the present work.

4. Analysis of Achieved Results

4.1. Qualitative Analysis of Generalization in Test Scenarios

After the training phase we tested the learned pedestrian model in some specific environments that were not “shown” to agents during the training, to understand if the approach was able to grant pedestrian a general capability to produce realistic behaviours even in newly encountered situations.

In particular, Figure 5 shows the “Anchor” environment, in which agents enter from the NE and NW corners, make a sharp bend and move North (a movement patten with a junction between two flows that is not that different from the T Junction environment): agents do not have particular problems, although they might have an hesitation close to the point in which the flows merge, due to the interaction and coordination process that must take place between pedestrians coming from the two entrances (something that is also plausible and that can be qualitatively observed in real world experiments).

Figure 6 shows the “Omega” environment, a maze-like structure in which 90° and u-turns to the right and to the left are present without choices among different passages. We emphasize that the training environments do not include all of these configurations for bends. Trained agents exhibit a reasonable behaviour, slowing down before the bends to avoid collisions with

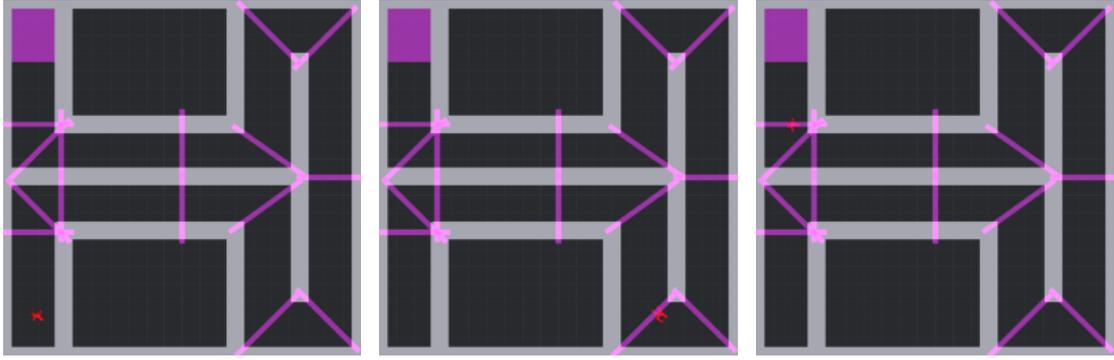


Figure 6: Omega environment execution.

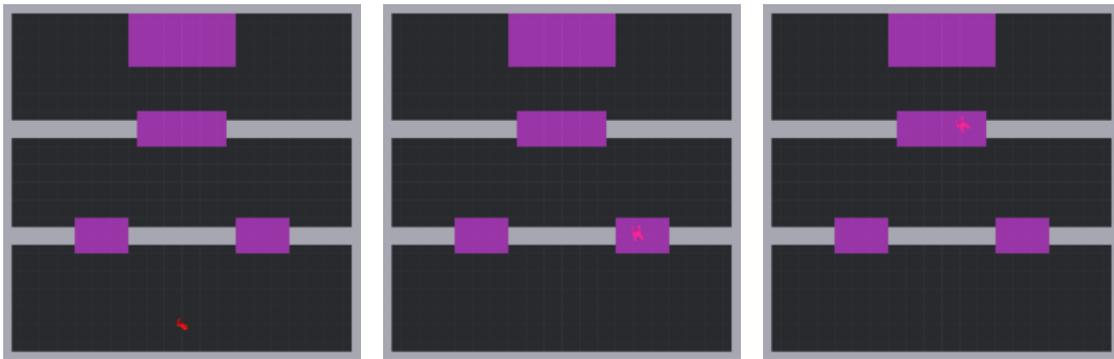


Figure 7: “Door choice” environment execution.

walls.

Figure 7 shows the “*Door choice*” environment, a relatively simple situation that however includes the choice of a passage from the Southern to the central region, in addition to a single passage to the Northern region that includes the final target. Within the training environments agents never face a situation in which they have to choose among two or more intermediate targets, and we wanted to find out if this kind of situation would instead be necessarily included in a proper curriculum for training pedestrian dynamics.

Trained agents actually do not have a problem in performing a plausible movement pattern in this scenario: they do not always choose the closest passage, but (i) real world experiments show that real pedestrians are not necessarily optimizing the expected travel time (although this generally happens when additional factors to distance, such as congestion, influence their decisions), and (ii) additional modifications to the model and to the training curriculum would be necessary to improve wayfinding behaviour to be competitive with hand-written and calibrated models.

Figure 8 finally shows the “*Bidirectional Door*” environment, a variant of the “Crowded Bidirectional Door” employed in the training. The lower number of pedestrians from the region beyond the passage, and their random initial position, paradoxically can represent a problem

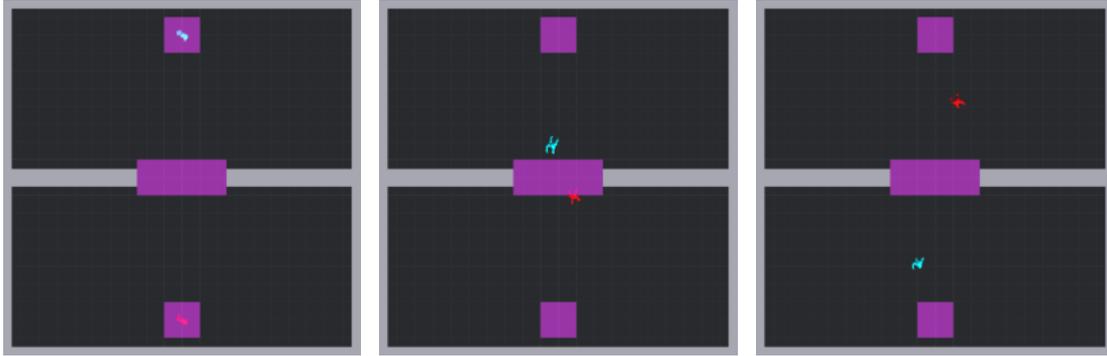


Figure 8: “Bidirectional Door” environment execution.

for the agents, since they cannot perceive the potential conflict until the very last moments. This scenario was therefore aimed at finding out if the trained agents were able to move at free-flow speed and then slow down when they perceive a conflicting pedestrian, avoiding it and, at the same time, don not completely disrupt the trajectory.

When agents had initial positions granting them immediate mutual perception they would start moving cautiously, and they cross the door keeping their right, then move to the final target. Otherwise, agents start moving at full velocity until they perceive each other, slowing down, and again change position to avoid each other when passing through the door, generally keeping their right. Sometimes agents do not follow the most direct path to the final target after passing through the door, but the overall behaviour is acceptable.

We did not test if the side preference is random and due to the randomness in the training process, or if there is some systematic bias (maybe due to the spatial structure of the training environments) that leads to an uneven distribution of this preference.

5. Conclusions and Future Developments

The paper has presented a research effort aimed at experimenting the adequacy of applying RL techniques to pedestrian simulation, especially considering the need to achieve general models applicable to a wide range of situations without the need of performing a training for each analyzed scenario. The achieved results are promising and encouraging. There are, of course, several limits of the current state of the research, representing lines for future research:

- we did not show a quantitative analysis of the achieved results, also for sake of space: this analysis, representing a first step in the direction of model validation, is object of current and future works;
- we intend to release as an open source project the developed software and the environments used for the curriculum and for the tests; at the moment of submission of the camera ready the repository is not ready, especially due to the lack of proper documentation, but anyone interested in accessing it can contact the authors (and plausibly future works on this line of research will include a reference to the accessible software repository);

- analysis of the effects of changes in RL algorithm, hyperparameters, configuration of the curriculum: we reached the presented solution performing some comparisons with alternative settings, but a systematic analysis of each of these aspect would require a focused specific work;
- additional quantitative experiments to improve the evaluation of the achieved results on the side of pedestrian simulation, towards a validation of the model or the acquisition of new objectives for model improvement;
- overcoming some current limits: modeling groups within the simulated pedestrian population is not possible, and preliminary work in this direction suggests that a change in the adopted RL algorithm would be necessary, due to the more systematic presence of agent to agent interaction; dealing with high density situations; going deeper in the capability of the model to perform wayfinding, possibly achieving the capability to adapt to the perceived level of congestion [19].

References

- [1] R. S. Sutton, A. G. Barto, Reinforcement Learning, an Introduction (Second Edition), MIT Press, 2018.
- [2] S. J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach (4th ed.), Pearson, 2020.
- [3] S. Bandini, S. Manzoni, G. Vizzari, Agent based modeling and simulation: An informatics perspective, *Journal of Artificial Societies and Social Simulation* 12 (2009) 4.
- [4] F. Martinez-Gil, M. Lozano, F. Fernández, Emergent behaviors and scalability for multi-agent reinforcement learning-based pedestrian models, *Simulation Modelling Practice and Theory* 74 (2017) 117–133. URL: <https://www.sciencedirect.com/science/article/pii/S1569190X17300503>. doi:<https://doi.org/10.1016/j.simpat.2017.03.003>.
- [5] N. Habbash, F. Bottoni, G. Vizzari, Reinforcement learning for autonomous agents exploring environments: an experimental framework and preliminary results, in: R. Calegari, G. Ciatto, E. Denti, A. Omicini, G. Sartor (Eds.), *Proceedings of the Workshop on 21st Workshop "From Objects to Agents"*, Bologna, Italy, September 14–16, 2020, volume 2706 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 84–100. URL: <http://ceur-ws.org/Vol-2706/paper5.pdf>.
- [6] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, Association for Computing Machinery, New York, NY, USA, 2009, pp. 41–48. URL: <https://doi.org/10.1145/1553374.1553380>. doi:10.1145/1553374.1553380.
- [7] L. Crociani, G. Vizzari, S. Bandini, Modeling environmental operative elements in agent-based pedestrian simulation, *Collective Dynamics* 5 (2020) 508–511. URL: <https://collective-dynamics.eu/index.php/cod/article/view/A85>. doi:10.17815/CD.2020.85.
- [8] L. Crociani, K. Shimura, G. Vizzari, S. Bandini, Simulating pedestrian dynamics in corners and bends: A floor field approach, in: G. Mauri, S. El Yacoubi, A. Dennunzio, K. Nishinari, L. Manzoni (Eds.), *Cellular Automata*, Springer International Publishing, Cham, 2018, pp. 460–469.

- [9] S. Paris, S. Donikian, Activity-driven populace: A cognitive approach to crowd simulation, *IEEE Computer Graphics and Applications* 29 (2009) 34–43. doi:10.1109/MCG.2009.58.
- [10] M. Haghani, M. Sarvi, Imitative (herd) behaviour in direction decision-making hinders efficiency of crowd evacuation processes, *Safety Science* 114 (2019) 49–60. URL: <https://www.sciencedirect.com/science/article/pii/S0925753518309275>. doi:<https://doi.org/10.1016/j.ssci.2018.12.026>.
- [11] E. T. Hall, *The hidden dimension*, Doubleday New York Ed., 1966.
- [12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, *CoRR abs/1707.06347* (2017). URL: <http://arxiv.org/abs/1707.06347>. arXiv:1707.06347.
- [13] F. L. D. Silva, A. H. R. Costa, A survey on transfer learning for multiagent reinforcement learning systems, *Journal of Artificial Intelligence Research* 64 (2019) 645–703. doi:10.1613/jair.1.11396.
- [14] B. Baker, I. Kanitscheider, T. M. Markov, Y. Wu, G. Powell, B. McGrew, I. Mordatch, Emergent tool use from multi-agent autotutorials, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020. URL: <https://openreview.net/forum?id=SkxpxJBKwS>.
- [15] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 06 (1998) 107–116. URL: <https://doi.org/10.1142/S0218488598000094>. doi:10.1142/S0218488598000094. arXiv:<https://doi.org/10.1142/S0218488598000094>.
- [16] J. Zhang, A. Seyfried, Comparison of intersecting pedestrian flows based on experiments, *Physica A: Statistical Mechanics and its Applications* 405 (2014) 316–325. URL: <https://www.sciencedirect.com/science/article/pii/S0378437114001988>. doi:<https://doi.org/10.1016/j.physa.2014.03.004>.
- [17] J. Zhang, W. Klingsch, A. Schadschneider, A. Seyfried, Transitions in pedestrian fundamental diagrams of straight corridors and t-junctions, *Journal of Statistical Mechanics: Theory and Experiment* 2011 (2011) P06004. URL: <https://doi.org/10.1088/1742-5468/2011/06/p06004>. doi:10.1088/1742-5468/2011/06/p06004.
- [18] D. Pathak, P. Agrawal, A. A. Efros, T. Darrell, Curiosity-driven exploration by self-supervised prediction, in: D. Precup, Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, PMLR, 2017, pp. 2778–2787. URL: <http://proceedings.mlr.press/v70/pathak17a.html>.
- [19] G. Vizzari, L. Crociani, S. Bandini, An agent-based model for plausible wayfinding in pedestrian simulation, *Engineering Applications of Artificial Intelligence* 87 (2020) 103241. URL: <https://www.sciencedirect.com/science/article/pii/S0952197619302246>. doi:<https://doi.org/10.1016/j.engappai.2019.103241>.

Information Seeking Behavior at the Time of COVID-19

Rino Falcone, Alessandro Sapienza

Institute of Cognitive Sciences and Technologies, ISTC-CNR,00185 Rome, Italy

Abstract

Italy was the first European country to be affected by COVID-19, facing an unprecedented situation. The reaction required drastic solutions and highly restrictive measures, which severely tested the trust of the Italian people. In this context, the role of information sources was fundamental, since they strongly influence public opinion. The central focus of this research is to assess how the information seeking behavior (ISB) of the Italian citizens affected their perception of government response strategies during the pandemic. Starting from the result of a survey addressed to 4260 Italian citizens, we used social simulation to estimate the evolution of public opinion. Particular attention has been given to different social categories, identified by age and gender. Comparing the ISB during and before COVID-19, we discovered that the shift in the ISB, during the pandemic, may have actually positively influenced public opinion, facilitating the acceptance of the costly restrictions introduced.

Keywords

CODIV-19, SARS-CoV-2, trust, fake news, misinformation, information-seeking behavior, social simulation

1. Introduction

The World Health Organization declared the COVID-19 situation as a pandemic on March 11, 2020. At that time, Italy was the European country most severely hit by this public health emergency. The COVID-19 outbreak forced the Italian State to deal with a novel, ambiguous, and unexpected risk. Politicians, local governments, and citizens had to face an unprecedented situation. The reaction to COVID-19 in Italy required drastic solutions and highly restrictive measures [19].

In this context of uncertainty and constant change, a strong need for information emerged [6], in order to understand what was going on: how the pandemic was evolving; to assess the actual risk to which all of us were subjected; to know community-level policies or personal health strategies. Quarantine, social distancing, mass swab tests, school closures, and the use of personal protective equipment are just some of the highly restrictive measures adopted to deal with the pandemic. People were required to undertake costly behavioral changes, not only in economic terms, but even restricting their personal freedom. All this was accompanied by a strong sense of uncertainty about the future. As Gualano and colleagues state [20], the Italian general population reported a high prevalence of mental health issues during the

WOA 2021: Workshop “From Objects to Agents”, September 1–3, 2021, Bologna, Italy

✉ rino.falcone@istc.cnr.it (R. Falcone); alessandro.sapienza@istc.cnr.it (A. Sapienza)

ORCID 0000-0002-0300-458X (R. Falcone); 0000-0001-7116-9338 (A. Sapienza)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

lockdown (depression, anxiety, poor sleep), whose impact is expected to persist beyond this critical situation.

In a time of crisis, the Italian population decided to trust its institutions and to rely on them to face COVID-19. This scenario was anything but trivial, considering the historical distrust in the Italian government and institutions in general. For sure, it is unthinkable that the authorities have suddenly become more reliable, but rather the citizens were forced by the circumstances. They had no other choice but to rely on their institutions, to trust the only entity able to face the problem, accepting all the necessary rules and restrictions [15].

Nevertheless, such a necessary choice may have resulted in a series of adjustments, in order to compensate and justify this "trust gap" [15]. As it is well known in the literature [9], in such cases, feedback and control mechanisms come into play. When there is not enough trust, making use of some type of control on the trustee allows one to lower the level of trust needed for reliance. From this consideration followed the great needs to get information, to monitor the institutions, to know that trust was well placed, and that the sacrifices the citizens were forced to make were fundamental. As Siegrist and Zingg report [41] in their systematic review on the importance of trust when preparing for and during a pandemic, trust is fundamental to positively influencing people's willingness to adopt recommended behavior.

In these cases, trust in institutions is therefore strongly linked to their communication skills, and to the fact that they are able to demonstrate the effectiveness of the proposed strategies. Even the World Health Organization, in a 2011 report [42], has identified communication as one of the biggest challenges, among the essential instruments required to tackle a pandemic. Our own opinions and beliefs are strictly tied to information we receive, especially in a moment of total uncertainty [3, 4, 40]; consider the lack of prior knowledge about this virus. This becomes particularly critical when, on the one hand there is an enormous risk, and on the other there is even the limitation of one's personal freedoms. Our trust is strongly related to information we receive and a lack of trust could lead to the non-acceptance of the rules imposed, the actual result of which depends not on the acceptance of the individual, but on the adherence of a substantial part of the population. In such a scenario, it is clear how fundamental it is to inform citizens properly, by identifying the right communication methods and limiting the spread of misinformation. As an example, manifold fake news [37] about COVID-19 were spread, probably aiming to affect public health communication and diminish preventive measures. The main, but not exclusive source of this fake news was social media. In order to answer this situation, the Italian Ministry of Health was forced to change its communication strategies, playing a strategic role in using its official Facebook page to mitigate the spread of misinformation and to offer updates to the online public [31]. Overall, the authorities' response to fake news was effective.

Another problem that arose, with respect to information, concerned the not always complete consistency of the indications provided by official sources (the various experts: virologists, infectious disease specialists, pandemic phenomena experts, etc.) as the pandemic proceeded. This phenomenon is well known to science, as many problems typically require a certain amount of time to be studied and to produce stable knowledge. Nevertheless, the population is not accustomed to such processes. Due to constant media pressure, motivated by the necessity to investigate the different perspectives of the pandemic, there has been an overexposure of experts in public debates. In several cases, citizens were faced with rapidly changing hypotheses, which

at times were also contradictory to each other. (Has the virus lost spreading and infectious capabilities or not? Can the asymptomatic subjects transmit the virus or not?)

The study presented in this paper contributes to this fast-growing body of knowledge on the interplay between trust in institutions and the COVID-19 pandemic, aiming to assess how the citizens' information-seeking behavior influenced the perception of government response strategies during the pandemic. Specifically, we exploited the result of a survey addressed to 4260 Italian citizens to determine their ISB. Then, we realized an agent-based simulation to study the evolution of public opinion, according to citizens' ISB.

We intend to test the hypothesis that the Italian population has behaved in a virtuous way, precisely because of the use they made of information. We believe that their virtuous behavior may have affected public opinion, by making it move towards compliance with the restrictive rules, which in turn may have reduced the impact of the pandemic.

2. Related Work

Information is a primary good for human beings, as it allows to reduce uncertainty and to make the world more predictable [7]. Our own opinions and beliefs are strictly tied to information we receive, especially if we are forced to face phenomena that are currently completely unknown [3, 4, 40]. Quality information represents the basis of good decision-making, which becomes pivotal when it comes to healthcare. Information seeking behavior (ISB) refers to those activities a person engages in when identifying his or her own need for information, searching for such information in any way, and using or transferring that information [21]. Health information seeking relates to the ways in which individuals obtain information, including information about their health, health promotion activities, risks to one's health, and illness [27]. As McCloud and colleagues state [33], the breadth and nature of health information obtained influences the individual's knowledge, beliefs, and attitudes toward a specific health behavior.

In the light of such considerations, the fundamental role of ISB is clear. Much attention in the literature has focused on identifying who actively seeks or who does not seek health information, the frequency of use, and satisfaction with health information seeking [23, 26]. The issue to date is that not all individuals seek health information equally.

Several studies have shed the light on the importance of demographic features for understanding information seeking, such as socioeconomic and ethnic diversity [36, 28]. Specifically, for what it concerns COVID-19, age and gender effects could be particularly interesting, given that they also represent the two main factors that determine the mortality for COVID-19 [13, 34, 35], together with the presence of pre-existing diseases.

As far as it concerns gender, many works underlined its strong effect on ISB. As Halder and colleagues [21] state, "Gender as a variable may be useful for better understanding the cognitive and social background of human information processing and may have important implications for information dissemination services and systems." The same authors, in their study, confirmed one of the well-known gender effects in the literature. They discovered that females seem to be more ardent information searchers when compared to males, and that they

also have more information needs than males. Similarly, Manierre [32] found that females are more likely to look for health information, with respect to males. Such an effect also applies to online sources [38].

Much less known, however, are the effects of age, with particular reference to the elderly. Although there are some studies in the literature, most of them focus on their specific use of the Internet [14, 25]. Yet, having a clear picture of older adults' health information-seeking behavior has an evident and substantial practical value. Indeed, given that this social category is generally the most subject to health-related risks, understanding how the elderly relate to information would be of great help to minimizing the diffusion of poor or potentially threatening health information or improving the diffusion of useful health information [1].

In summary, given the great importance of the ISB in decision-making processes, it is clear that studying it is fundamental to understanding how individual citizens (and the whole community) responded to the COVID-19 emergency. These considerations would be particularly useful both for public institutions and for the healthcare system, allowing a better understanding of what happened and how public opinion could be better orientated in the future.

3. Survey and Sample

We will not report the full details of the entire study, which can be found at [15, 17, 18]. In this specific work, we focus our attention on a particular subsection of the original survey, taking a closer look at the information seeking behavior of the Italian citizens.

The study was conducted using a snowball sampling method to determine the respondents: it concerned a large sample ($N = 4260$, 57% women, mean age = 46 years), relatively well-balanced in terms of geographical provenance (33% Northern Italy, 39% Central Italy, 28% Southern Italy and main islands), with a significant portion of respondents (30%) residing in the regions most affected by COVID-19 at that time (Lombardy, Veneto, Emilia-Romagna, Marche, Piedmont). It should be noted that the mean educational level of participants was very high: almost three quarters of respondents had a degree (38%) or post-graduate specialization (34%). The main characteristics of the sample are shown in Table 1.

Data were collected with a 57-item questionnaire, using a 5-point Likert scale for most items. An English translation of the whole questionnaire is available at [15].

The questionnaire was aimed to investigate the participant's overall trust towards public authorities and their motivations, along with the factors that determine the participant's trust. The questionnaire was based on the socio-cognitive model of trust developed by Castelfranchi and Falcone [10] and explored participants' opinions on five main dimensions, in relation to the current COVID-19 crisis in Italy:

1. Evaluation of the competence of public institutions;
2. Evaluation of the intentionality of public institutions;
3. Purposes and effectiveness of the public institutions' intervention;
4. Trust and information sources: the most used sources of information and their perceived trustworthiness;
5. Expectations about the future scenarios that will arise, once the COVID-19 crisis is over.

Table 1
Sample characteristics.

	Regions Most Affected % (30%)	Regions Less Affected % (70%)	Total %
<i>Gender</i>			
Men	45	42	43
Women	55	58	57
Total	100	100	100
<i>Age (Mean = 46)</i>			
18–29	19	11	13
30–39	23	18	19
40–49	23	24	24
50–59	21	28	26
60–69	11	15	14
>70	3	4	4
Total	100	100	100
<i>Educational Level</i>			
Middle school	3	2	2
High school	24	27	26
University degree	41	36	38
Post-graduate specialization	32	35	34
Total	100	100	100
<i>Geographical provenance</i>			
Northern Italy	96	7	33
Central Italy	4	53	39
Southern Italy/islands	0	40	28
Total	100	100	100

As already stated, in this work we will mainly focus on the fourth point. Specifically, we analyzed the information seeking behavior (ISB) of the respondents, with particular reference to the following sources of information:

- Traditional media (TM);
- Official websites (OW);
- Social media (SM);
- Family physicians (FP);
- Scientists (S);
- Friends, relatives, acquaintances (FRA).

3.1. Effects of Age and Gender

As highlighted in Section 2, age and gender are two particularly interesting features, given their influence on ISB. Furthermore, in this specific domain, they strongly affect the mortality for COVID-19. This section has the purpose of presenting the peculiarity introduced by such features.

Comparing male and female respondents, significant differences emerged concerning their ISB. We identified an average 4% higher information request for women, with respect to men. Such a difference increases when we consider the use of online sources. There is an increment of +6.6%, confirmed by correlation data, both for official websites ($R = 0.124$, $p < 0.0001$) and social media ($R = 0.1$, $p < 0.0001$). This effect is well-supported by previous evidence in the literature, since other studies detected a higher tendency of women to refer to online sources for health information [8, 44]. This effect disappears when we consider scientists as information sources (there is a slight difference of -0.7%). On the contrary, we did not find significant gender effects about trust.

Investigating the effects of age, we also considered a 10-year range re-coding. The idea that led to the categorization process was to investigate the behavior of individuals subjected to the same level of risk (death rate). In this regard, we refer to the most common death rate classifications (age, sex, existing conditions of COVID-19 cases and deaths, <https://www.worldometers.info/coronavirus/coronavirus-age-sex-demographics/>; accessed at 15/03/2020) [35], considering age in 10-year ranges. As far as it concerns age, young people tend to consult information sources with a lower frequency, compared to older people, going from an average frequency of use of 53.86% for 18–29 to a 61.41% for people over 70. The different levels of risk that COVID-19 entails between young and old people [34, 35], in particular in the severity it can take, could at least in part explain the differences introduced by age regarding the willingness to keep informed about this phenomenon. Young people were also the most suspicious with respect to sources of information: 30–39 year-old people showed the lowest level of trust (57.71%, slightly higher than 18–29), which increased with age until 64.03% for people over 70.

Combining the effects of age and gender (Table 2), we found that 18–29 year-old men had the lowest average frequency of use (52.9%) while 30–39 year-old men had the lowest level of trust (58.9%). In contrast, over 70 year-old women were the most prone to use information sources (63.3%) and to trust them (68.2%).

Table 3 reports the frequency of use for the different information sources. Focusing on the least trusted ones, we observed that people over 70, more exactly women, are the ones that make most use of FRA and social media. The people over 70's average use of social media was 37.4%, and it even increased up to 49.2% for women. Advanced age does not seem to have represented a barrier to accessing social media. This result is even more striking when crossed with the analysis of trust (Table 4). The average trust in social media is about 20%, but it rises to 29% for men over 70, and even to 39% for women over 70. Summarizing, on the one hand, women over 70 make extensive use of information sources; on the other hand they show a higher level of trust in social media, which certainly cannot be considered the most reliable source. A deeper analysis of these two contradictory characteristics may offer some interesting insights. Given the critical role age and gender have on ISB, we decided to introduce them in our framework as categorization criteria. This allowed us to understand how the different social categories use information sources and what impact this has on their opinion. Combining age and gender, we considered a total of 12 social categories.

Table 2

Average values of frequency of use and trust in information sources, based on age and gender.

Category	Average Use	Average Trust
women 18–29	55.36 %	60.85%
women 30–39	56.54%	60.02%
women 40–49	59.10%	61.30%
women 50–59	59.58 %	61.99%
women 60–69	61.83%	63.27%
women over 70	63.33 %	68.15%
men 18–29	51.93%	59.85%
men 30–39	52.43%	58.86%
men 40–49	54.07%	59.94%
men 50–59	55.46%	59.78%
men 60–69	57.85%	60.68%
men over 70	60.01%	63.11%

4. Simulations

In this work, the use of simulations has allowed us to estimate how the opinions of classes of individuals and of a whole community evolved, starting from the specific use the citizens made of information sources. In order to study opinion dynamics, we considered a model based on that proposed by Hegselmann–Krause [22], whose effectiveness has been proven over time. Their model has been modified here by introducing the probabilistic use of information sources.

Then, given the belief b as "the institutional truth" to tackle COVID-19 outbreak (i.e., "it is essential to use face masks," or "it is necessary to maintain social distance"), the citizens, modeled in the simulation as agents, had different opinions about said belief, since the information sources they used may have reported evidence supporting b or opposing it. As we supposed b to be the institutional truth, opposing information represents, for simplicity, misinformation and/or fake news.

Let n be the number of information sources under consideration. To model the repeated process of opinion formation, we considered a discrete-time system; thus time was divided in rounds $T = \{0, 1, 2, \dots\}$. For a fixed agent i , we denote its opinion at time t by $x_i(t)$, expressed as a real number in $[0,1]$. Similarly, for a fixed information source j , we denote information of the sources j at time t by $inf_j(t)$. While fixing an agent i , δ_i represents the weight i gives to its own opinion, and a_{ij} is the weight given to information coming from the source j , with $1 \leq j \leq n$.

Furthermore, let $\delta_i \geq 0$ and $a_{ij} \geq 0$ for all i and j , and let their sum be equal to 1, as in Equation (2). This last condition is not mandatory, but it allows us to avoid the normalization process. Considering this notation, opinion formation of agent i can be described as in Equation (1).

$$x_i(t+1) = \delta_i * x_i(t) + a_{i1} * inf_1(t) + a_{i2} * inf_2(t) + \dots + a_{in} * inf_n(t) \quad (1)$$

$$\delta_i + a_{i1} + a_{i2} + \dots + a_{in} = 1 \quad (2)$$

In addition to the Hegselmann–Krause model, we considered that i would not use all its source at each round, but just a subset of them. This characteristic allowed us to model a frequency-based

access to the sources, which is precisely what the citizens stated to do. Therefore, we introduce the function $use_{ij}(t)$, whose Boolean result determines if i will make use of the source j at time t . Thus, our model will be described by Equations (3) and (4).

$$x_i(t+1) = \delta_i * x_i(t) + a_{i1} * inf_1(t) * use_{i1}(t) + a_{i2} * inf_2(t) * use_{i2}(t) + \dots + a_{in} * inf_n(t) * use_{in}(t) \quad (3)$$

$$\delta_i + a_{i1} * use_{i1}(t) + a_{i2} * use_{i2}(t) + \dots + a_{in} * use_{in}(t) = 1 \quad (4)$$

Given this framework, we are interested in understanding how the opinion of the agent i , characterized by an initial profile and a specific ISB, evolves over time. By doing this analysis for each category of citizens, we are able to determine how the particular ISB of a category influences the final opinion of the citizens belonging to that category.

The data we need are:

1. Initial profile of the agent i , given by $x_i(0)$ and δ_i ;
2. ISB, i.e., $use_{ij}(t)$ and a_{ij} for each information source j ;
3. Actual trustworthiness of each information source j , in order to generate $inf_j(t)$.

As far as it concerns $x_i(0)$ and δ_i , we considered a situation in which these two parameters did not affect the final outcome. Notice that, in such a system, given that information sources influence citizens but the opposite never happens, after a sufficiently wide time window $x_i(t)$ can be considered as independent of $x_i(0)$. Accordingly, we introduced a long transient phase, which ensured that this condition holds. With respect to δ_i , it affects the stability of $x_i(t)$ in time, determining how much it can vary from round to round: increasing δ_i makes $x_i(t)$ more stable and vice versa. In order to nullify the influence of δ_i , we did not simply analyze the final value of $x_i(t)$, but its average value after the transient phase.

The ISB is definitely the most interesting part, since in this experiment we exploited it to study its effect on the citizens' opinion. In such a context, $use_{ij}(t)$ is represented by the frequency of use of the sources. For instance, if i has a frequency of use of 50% for source j , this means that there is the 50% probability that i will access this source at time t . For what it concerns the weights given to the reported information, a_{ij} , this can be generated by the trust value i has on j for reporting information about COVID-19. For the sake of simplicity, we used already normalized weights, as in Equation (4) (of course, given Equation (4), it is also necessary that $\delta_i \neq 1$; otherwise, $x_i(t)$ will not be affected by information sources). Specifically, we keep δ_i fixed, and we assign the weights of the other sources proportionally to their trust values. The choice to use trust values as weights for information source has a solid foundation in the literature [2, 5, 16, 39] for other models and specifically for Hegselmann–Krause [24]. All the data we need to generate $use_{ij}(t)$ and a_{ij} , the frequency of use and the average trust, can be found respectively in Tables 3 and 4.

We come therefore to the last parameter of the experimental setting, the actual trustworthiness of the information sources. Unfortunately, there is no way to find such data. To the best of our knowledge, we could not find any study about it, which is reasonable, since it is not that easy to produce a precise quantification of how trustworthy these information sources were during the pandemic. Nevertheless, two possible approaches allowed us to overcome the problem. The first one consists of equalizing trust (the assessment of the trustor) and trustworthiness (the intrinsic

Table 3

Frequency of use for each information sources, based on age and gender.

Category	TM	OW	SM	FP	Scientists	FRA
women 18–29	77.82%	78.61%	42.24%	31.58%	59.56%	42.32%
women 30–39	75.67%	83.73%	41.94%	32.54%	67.08%	38.31%
women 40–49	82.33%	83.77%	39.92%	38.86%	74.53%	35.17%
women 50–59	84.75%	82.19%	37.81%	39.82%	76.93%	35.97%
women 60–69	87.38%	80.05%	38.88%	46.21%	81.31%	37.15%
women over 70	91.25%	61.25%	49.17%	48.33%	79.17%	50.83%
men 18–29	68.85%	75.81%	35.79%	25.20%	66.94%	39.01%
men 30–39	73.44%	78.28%	31.80%	26.39%	69.92%	34.75%
men 40–49	76.89%	77.24%	34.67%	30.84%	72.82%	31.96%
men 50–59	80.14%	74.79%	34.53%	36.76%	75.95%	30.56%
men 60–69	86.84%	70.94%	29.77%	47.88%	78.62%	33.04%
men over 70	88.41%	65.85%	35.98%	54.88%	75.00%	39.94%

Table 4

Trust in information sources, based on age and gender.

Category	TM	OW	SM	FP	Scientists	FRA
women 18–29	55.09%	89.73%	19.83%	64.66%	88.56%	29.78%
women 30–39	54.53%	87.88%	19.27%	64.79%	86.78%	30.34%
women 40–49	55.72%	87.29%	20.34%	69.32%	90.04%	30.04%
women 50–59	57.63%	86.64%	20.72%	69.54%	89.82%	32.63%
women 60–69	59.38%	86.04%	21.45%	71.77%	91.32%	34.70%
women over 70	65.42%	84.58%	38.75%	75.42%	90.42%	48.75%
men 18–29	51.01%	87.60%	20.87%	66.23%	88.51%	30.24%
men 30–39	50.33%	85.57%	16.39%	66.64%	87.38%	28.85%
men 40–49	55.25%	84.73%	18.51%	69.93%	87.38%	28.54%
men 50–59	56.46%	82.15%	18.33%	69.70%	88.56%	29.56%
men 60–69	59.19%	81.54%	19.43%	70.94%	89.13%	31.27%
men over 70	60.06%	77.44%	28.96%	73.17%	91.16%	39.33%

property of the trustee, determining its actual performance). Given that it is rarely possible to directly access the effective value of trustworthiness, we may reasonably suppose that the assessment of the survey respondents approximates sufficiently well the real trustworthiness of the information sources. To give a clearer idea, this is exactly what currently happens in systems such as Google, Ebay or TripAdvisor: the evaluation of a seller/restaurant/hotel is obtained by aggregating the evaluations produced by a multitude of individuals; this is not the real trustworthiness, but it is an appropriate approximation. In this specific case, we easily extrapolated such data from the questionnaire. We believe that this first approach (average trust value—*ATV*), with a certain margin of unavoidable error, represents the most precise solution.

As an alternative, it is possible to arbitrarily rank the sources on the basis of their perceived trustworthiness (trustworthiness-based sorting—*TBS*), assigning the specific values accordingly. In the *ATV* case, these values approximately vary from a minimum of 20% to a maximum of 90%. We consider the same ranking and the same range in the *TBS* case, distributing the 6 sources in intervals of equal dimensions. In this study, we took into account both the possibilities.

As already stated, the trustworthiness of a source determines $inf_j(t)$; i.e., it represents the actual probability that the source j will report information supporting the belief b , at time t . For instance, in the ATV case, scientists will report supporting information ($inf_j(t) = 1$) 88.85% of the time, and opposing information ($inf_j(t) = 0$) the rest of the time.

Considering the workflow of the simulation, at the beginning of each run, we found the various information sources, each described by a specific value of trustworthiness, and a number of citizens. Each citizen i belongs to a specific category, identified by age and gender, which in turn determines its ISB. At each round of this phase:

1. The sources report fresh information supporting b or opposing to it;
2. The citizen i accesses a subset of the available information sources, according to its ISB, and updates its opinion $x_i(t)$ according to the formula 3.

The simulation consists of two different parts: the transient phase, whose sole function is to ensure that the citizen's initial profile will not affect $x_i(t)$ in the next phase; the analysis phase, in which we actually analyze the value of $x_i(t)$. In particular, we are interested in the average value of $x_i(t)$ during this last phase, which is used, within this framework, as a performance indicator. Ideally (for the institutions), this dimension should tend towards the target value 1 (the citizen believes b to be true), while low values are undesirable (the citizen does not believe b).

The resulting model has been implemented in the simulation environment NetLogo (version 5.2) [43].

5. Results

In the following experiments we considered a social simulation by implementing the model introduced in the previous sections. Specifically, we were interested in:

- Determining how much the ISBs of the different categories of citizens, classified by age and gender, affected their opinions, and in turn, their choices during the pandemic.
- Trying to compare the ISBs identified in this study with those prior to COVID-19 arrival. Such a comparison may help determine whether and to what extent the citizens' rational and responsible choice to rely on trusted sources positively affected their acceptance of restrictions and rules needed to face the pandemic.

5.1. First Experiment: The Influence of ISBs on the Citizens' Opinions

The analysis in Section 3.1 suggests that women and older people show a greater interest in information (regarding the pandemic), while young men seem to be the most disinterested category. That being said, it is still necessary a more in-depth analysis to understand how these characteristics affected the opinion of the individuals, making them more or less resilient to misinformation. Especially for women over 70, we thought it interesting to verify how their frequent usage and high trust in social media have affected their opinions, and whether their higher use of the other sources can compensate for these contrasting characteristics. More generally, it is particularly useful to quantify the impacts of information sources on the various

Table 5

Trustworthiness of the sources of information.

source of information	ATV	TBS
Scientists	88.85%	90%
OW	85.82%	76%
FP	68.63%	62%
TM	55.91%	48%
FRA	31.04%	34%
SM	20.02%	20%

categories of citizens. Consider, for instance, that being able to study and to quantify the evolution of their opinion would allow the institutions to identify which part of the population needs targeted interventions, offering the possibility to optimize the available resources.

While aiming to quantify those dimensions, within this section we introduce an agent-based simulation, which allows us to evaluate how the citizens' opinion changes depending on their ISBs. In particular, we considered an opinion dynamics model based on that proposed by Hegselmann–Krause [22]. Their model is used to investigate opinion dynamics within a group of individuals who interact with each other by exchanging opinions. Within this experiment, we focus on the opinions of individuals, in a context in which they receive information from their sources in a unidirectional way (the source reports information to the individual, but the opposite does not happen). Conversely, in the next experiment we consider the opinion of the whole population.

We report below the experimental setting:

1. Transient phase: 2000 rounds. Actually, we verified that a significantly lower number of rounds would be sufficient. Yet, we decided to use a high value.
2. Analysis phase: 100 rounds.
3. $x_i(0)$: 0.5. As we stated, this parameter did not affect the final values.
4. δ_i : 100%. As we stated, this parameter did not affect the final values.
5. Number of citizens: 1200, i.e., 100 for each category.
6. Frequency of use in Table 3.
7. Trust in information sources in Table 4.
8. Trustworthiness of the information sources for the *ATV* and *TBS* cases in Table 5.

The data summarized in Table 6 show the findings of the experiment, both in the *ATV* case and the *TBS* case. Overall, we observe a positive tendency, as the average opinion of all the categories of citizens is over the threshold 0.5. This result was driven by the intensive use of reliable information sources, first, above all, scientists. Nevertheless, it is worth noting that in none of the cases analyzed did we encounter a striking value, particularly close to the target value 1.

Coming then to detail, it is plainly clear that women over 70 had the worst performance. Their average opinion is 0.661 in the *ATV* case and 0.628 in the *TBS* case.

In contrast, 30–39 year-old men showed the best performance: their average opinion was 0.731 in the *ATV* case (+7% with respect to the worst case) and 0.69 in the *TBS* case (+6.2% with respect to the worst case).

Table 6

Average values of the citizens' opinions during the analysis phase. The citizens are grouped by age and gender. We report average values over 100 simulations.

Category	Average Opinion—ATV	Average Opinion—TBS
Women 18–29	0.704	0.661
Women 30–39	0.719	0.677
Women 40–49	0.723	0.682
Women 50–59	0.723	0.683
Women 60–69	0.715	0.677
Women over 70	0.661	0.628
men 18–29	0.718	0.679
men 30–39	0.731	0.69
men 40–49	0.726	0.686
men 50–59	0.723	0.684
men 60–69	0.716	0.677
men over 70	0.69	0.654

5.2. Second Experiment: A Comparison between the Citizens' ISBs before and after the Arrival of COVID-19

All the data and the analyses previously reported in this work clearly suggest a substantial virtuous behavior of the Italian population. In particular, in the attempt to get reliable information, there has been an extraordinarily high reliance on science. We can then assert that this virtuous behavior actually played an important role in the fight against the pandemic: being properly informed encouraged the acceptance of the rules and guidelines, despite the high personal burden.

It would be interesting to quantify to which extent this virtuous behavior has helped to reduce the impact of COVID-19. In this sense, our effort within this second experiment was to compare the estimation of the public opinion in this case with what would have happened if the population had behaved, with respect to information, as it usually did in health-related contexts before COVID-19. We made use of the same framework of the previous experiment. If, however, we previously evaluated the individual categories, here we were interested in studying the opinion of the whole population. In other words, we implemented the logic of the previous experiment, considering a whole community, made up of citizens belonging to the same categories. In order to obtain a more likely outcome, we took into account the real distribution of the Italian population, according to age and gender. Such data, current as of January 2019, have been retrieved from the Istat website (<http://dati.istat.it/Index.aspx?QueryId=42869> accessed at 07/08/2020). Table 7 summarizes them.

We considered a population of 1000 citizens, which allowed for a more precise distribution of the citizens among the categories. Then we analyzed the average opinion of the whole population. We compared the results obtained with the ISBs extrapolated from our survey (outbreak setting—*OS*), concerning the COVID-19 outbreak, with the one (prior setting, *PS*) retrieved by the fairly recent study of Zucco and colleagues [44], conducted in May 2017 in Italy ($N = 913$). The study aimed to identify the ISB concerning antibiotics and health in general. Even

Table 7

Distribution of the Italian population by age and gender, current as of January 2019.

Category	Men	Women
18–29	7.55%	7.01%
30–39	7.02%	6.94%
40–49	9.06%	9.17%
50–59	9.03%	9.42%
60–69	6.93%	7.55%
over 70	8.55%	11.76%

Table 8

Frequency of the use of information sources extrapolated from the study of Zucco and colleagues.

	TM	OW	SM	FP	Scientists	FRA
frequency of use	13.6%	27.6%	45.1%	71.6%	8.9%	1.7%

if it does not provide all the data we needed concerning health in general, it does for antibiotics. Moreover, at least for what concerns the available data, the frequencies of use appear quite similar in these two contexts. Thus, we considered data about information sources' usage for antibiotics (Table 8). We considered the same population of 1000 citizens, verifying how the average opinion changed when introducing the ISBs of Italian population before COVID-19.

Summarizing the experimental setting:

1. Transient phase: 2000 rounds. Actually, we verified that a significantly lower number of rounds would be sufficient. Yet, we decided to use a high value.
2. Analysis phase: 100 rounds.
3. $x_i(0)$: 0.5. As we stated, this parameter did not affect the final values.
4. δ_i : 100%. As we stated, this parameter did not affect the final values.
5. Number of citizens: 1000, distributed by age and gender according to Table 7.
6. Frequency of use: Table 3 for OS agents and Table 8 for PS agents.
7. Trust in information sources in Table 4.
8. Trustworthiness of the information sources in Table 5.

Results in Table 9 highlight a substantial difference of opinion. Specifically, OS agents had +8.2% and +9.1% higher outcomes, respectively, in the ATV and TBS cases. These findings suggest that the responsible behavior of the population, with respect to information, has indeed helped the acceptance of the rules imposed by the institutions and it may have contributed to reducing the impact of COVID-19.

6. Discussion

In this study, we focused on the ISB of the Italian population during the early stages of the pandemic. In a situation of high risk and uncertainty, accurate information becomes priceless. This is what happened in Italy, in the early stages of the COVID-19 pandemic. On the one hand,

Table 9

Average opinion of the OS and PS populations, in the ATV and TBS cases.

	Average Opinion—ATV	Average Opinion—TBS
<i>OS</i>	0.71	0.664
<i>PS</i>	0.628	0.573

there was the growing need to be informed about what was happening in order to understand the general characteristics, the risks, and the evolution of the phenomenon. On the other hand, information represented a way to check and control the actions of the institutions, to check that everyone's personal sacrifice was not in vain, that the granted trust was not misplaced. Even the World Health Organization, in a 2011 report [42], identified communication as one of the biggest challenges to tackle a pandemic. We found confirmation of this in our analysis, detecting a strong correlation ($R = 0.545$, $p < 0.0001$) between trusting institutions for managing the COVID-19 emergency and as a source of information for the same topic.

From the need to rely on the institutions, in a historical context of substantial distrust, followed the necessity to introduce control mechanisms to fill this trust gap, to supervise somehow the actions of the institutions. What we detected here is a well known phenomenon in the literature [9, 11]. When the trustor (citizens) does not trust enough the trustee (Italian institutions), it can decide to introduce control mechanisms to check the trustee's actions. Thus, the trustor (to realize the practical act of trusting) needs a lower level of trust (as mental attitude). In our specific case, to trust the institution was the only possible choice; thus, the higher information request [6] was introduced as a control mechanism.

This affected not only the quantity, but also the quality of the information requested. Indeed, we detected significant changes with respect to people's behavior towards information. Such changes may have positively affected the effectiveness of the measures introduced to tackle the pandemic, facilitating their acceptance by the whole population. The most remarkable result is probably the very high usage of scientists as information sources. About 92.6% of the respondents reported to trust scientists. This is definitely far from what has been found in other studies [44], especially if we consider the marginal role science has played in Italy over the last few decades. It is worth noting that, before COVID-19, Italy had a fairly worrying scientific picture, both from the political and the societal perspective (decreasing funding for the sector, and citizens' distrust towards scientific rationality: think of phenomena such as flat-Earthers or anti-vaxers). After the COVID-19 outbreak, science has suddenly recovered an important role in the citizens' consideration. This is quite reasonable because the only concrete answer (both as analysis of the phenomenon and as ability to mitigate and oppose to it by new tools and approaches) was the one provided by science and scientific research.

As far as it concerns age and gender, we found that they significantly affect the ISB. For example, it is interesting to note the case of women over 70 compared to other categories: they make a particularly relevant use of social media. This is alarming, since the category subjected to the higher risk is also the one that relies most on those that are widely identified as the least reliable media. Experiments confirm that this characteristic has a strong negative impact on women over 70, such that their average opinion is the lowest ever (compared to the institutional point

of view). It is a serious problem that one of the groups subjected to the highest risk [12, 30] relies that much on what looks like the worst information source (at least from the institution's point of view). Therefore, such a negative performance is not related to lack of information, but rather to a lower ability to relate to information [29]: they probably do not know they should distrust social media.

The most interesting and remarkable contribution of this work is the comparison of the ISB detected during the pandemic with that detected in other studies before the arrival of COVID-19. Indeed, it appears that the exceptionally responsible use of information sources has had a positive effect on the opinion of the Italian population (+8.2% and +9.1% higher, respectively in the ATV and TBS cases). Such a difference is particularly critical, as it concerns the average opinion of the population. Consider, in fact, that the effective success of the rules introduced to deal with COVID-19 depended on the adoption of these rules by a significant percentage of the Italian population. Therefore, a lower average opinion was not just a symptom of a lower acceptance level, but it could have involved a chain reaction, discouraging even those who considered such measures useful. Such a situation was clear both to citizens and institutions: the perceived utility of these measures positively correlated with believing that a sufficient number of individuals will follow these restrictions ($R = 0.233$, $p < 0.0001$) and negatively correlated with believing that an insufficient number of people will do that ($R = -0.21$, $p < 0.0001$). In view of these findings, we may conclude that the responsible behavior we detected in this study, with respect to information, has indeed helped the acceptance of the strict and restrictive rules imposed by the institutions and it may have contributed to reducing the impact of COVID-19.

7. Conclusions

In this work, we investigated the relationship between COVID-19 and the citizens' need for information. This need for information explains the disproportionate increase in trust that we have witnessed (and that the original survey [15] accurately reported). In fact, a verification and control mechanism came into play, acting as a compendium (and as a rational and non-fideistic moderator) to the disproportionate trust we have witnessed (and which proved to be necessary to face such a delicate phase). Verifying and controlling the development of the epidemic and the countermeasures put in place with their potential effects and findings represented a particular input that is clearly captured by this study which defines its characteristics and methods.

This study shows how the role of information was fundamental in dealing with COVID-19. Thanks to the possibility of obtaining information and comparing the different sources of information, citizens have been able to appreciate the fundamental role played by science in modern societies. Obviously, even the information coming from scientists has had a dynamic (with some contradictory positions) strictly linked to the awareness that science has acquired over time on the phenomenon in question. Our hope is to witness a maturation of public opinion towards the sources of information that persist over time. Moreover, although the critical phase of the pandemic seems to be over, it is essential to continue monitoring its evolution, taking care of the complex relationship between the institutions and citizens. We sincerely hope that the considerations included in this work, in addition to clarifying what has happened in the past months and what has worked or not, can also help public institutions and the healthcare

system in the future.

References

- [1] Altizer, K. P., Grzywacz, J. G., Quandt, S. A., Bell, R., & Arcury, T. A. (2014). A qualitative analysis of how elders seek and disseminate health information. *Gerontology & geriatrics education*, 35(4), 337-353.
- [2] Amgoud, L., & Demolombe, R. (2014). An argumentation-based approach for reasoning about trust in information sources. *Argument & Computation*, 5(2-3), 191-215.
- [3] Austin, L., Fisher Liu, B., & Jin, Y. (2012). How audiences seek out crisis information: Exploring the social-mediated crisis communication model. *Journal of applied communication research*, 40(2), 188-207.
- [4] Austin, L., & Jin, Y. (2016). Social media and crisis communication: Explicating the social-mediated crisis communication model. In *Strategic Communication* (pp. 175-198). Routledge.
- [5] Barber, K. S., & Kim, J. (2001). Belief revision process based on trust: Agents evaluating reputation of information sources. In *Trust in Cyber-societies* (pp. 73-82). Springer, Berlin, Heidelberg.
- [6] Bento, A. I., Nguyen, T., Wing, C., Lozano-Rojas, F., Ahn, Y. Y., & Simon, K. (2020). Evidence from internet search data shows information-seeking responses to news of local COVID-19 cases. *Proceedings of the National Academy of Sciences*, 117(21), 11220-11222
- [7] Berger, C. R., & Calabrese, R. J. (1974). Some explorations in initial interaction and beyond: Toward a developmental theory of interpersonal communication. *Human communication research*, 1(2), 99-112.
- [8] Brodie, M., Flournoy, R. E., Altman, D. E., Blendon, R. J., Benson, J. M., & Rosenbaum, M. D. (2000). Health Information, The Internet, And The Digital Divide. *Health affairs*, 19(6), 255-265.
- [9] Castelfranchi, C., & Falcone, R. (2000). Trust and control: A dialectic link. *Applied Artificial Intelligence*, 14(8), 799-823.
- [10] Castelfranchi, C., & Falcone, R. (2010). *Trust theory: A socio-cognitive and computational model* (Vol. 18). John Wiley & Sons.
- [11] Das, T. K., & Teng, B. S. (1998). Between trust and control: Developing confidence in partner cooperation in alliances. *Academy of management review*, 23(3), 491-512.
- [12] Davies, N. G., Klepac, P., Liu, Y., Prem, K., Jit, M., Eggo, R. M., & CMMID COVID-19 working group. (2020). Age-dependent effects in the transmission and control of COVID-19 epidemics. *MedRxiv*.
- [13] Dowd, J. B., Andriano, L., Brazel, D. M., Rotondi, V., Block, P., Ding, X., ... & Mills, M. C. (2020). Demographic science aids in understanding the spread and fatality rates of COVID-19. *Proceedings of the National Academy of Sciences*, 117(18), 9696-9698.
- [14] Escoffery, C. (2018). Gender similarities and differences for e-Health behaviors among US adults. *Telemedicine and e-Health*, 24(5), 335-343.
- [15] Falcone, R., Coli, E., Felletti, S., Sapienza, A., Castelfranchi, C. & Paglieri, F. (In press).

All we need is trust: How the COVID-19 outbreak reconfigured trust in Italian public institutions. *Frontiers in Psychology*.

- [16] Falcone, R., Sapienza, A., & Castelfranchi, C. (2015). The relevance of categories for trusting information sources. *ACM Transactions on Internet Technology (TOIT)*, 15(4), 1-21.
- [17] Falcone, R., & Sapienza, A. (2020). How COVID-19 Changed the Information Needs of Italian Citizens. *International Journal of Environmental Research and Public Health*, 17(19), 6988.
- [18] Falcone, R., & Sapienza, A. (2021). An agent-based model to assess citizens' acceptance of COVID-19 restrictions. *Journal of Simulation*, 1-15.
- [19] Grasselli, G., Pesenti, A., & Cecconi, M. (2020). Critical care utilization for the COVID-19 outbreak in Lombardy, Italy: early experience and forecast during an emergency response. *Jama*, 323(16), 1545-1546.
- [20] Gualano, M.R.; Lo Moro, G.; Voglino, G.; Bert, F.; Siliquini, R. Effects of Covid-19 Lockdown on Mental Health and Sleep Disturbances in Italy. *Int. J. Environ. Res. Public Health* 2020, 17, 4779.
- [21] Halder, S., Ray, A., & Chakrabarty, P. K. (2010). Gender differences in information seeking behavior in three universities in West Bengal, India. *The International Information & Library Review*, 42(4), 242-251.
- [22] Hegselmann, R., & Krause, U. (2002). Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of artificial societies and social simulation*, 5(3).
- [23] Hesse, B. W., Nelson, D. E., Kreps, G. L., Croyle, R. T., Arora, N. K., Rimer, B. K., & Viswanath, K. (2005). Trust and sources of health information: the impact of the Internet and its implications for health care providers: findings from the first Health Information National Trends Survey. *Archives of internal medicine*, 165(22), 2618-2624.
- [24] Ishii, A. (2019, June). Opinion dynamics theory considering trust and suspicion in human relations. In *International Conference on Group Decision and Negotiation* (pp. 193-204). Springer, Cham.
- [25] Jacobs, W., Amuta, A. O., & Jeon, K. C. (2017). Health information seeking in the digital age: An analysis of health information seeking behavior among US adults. *Cogent Social Sciences*, 3(1), 1302785.
- [26] Kirschning, S., & von Kardorff, E. (2008). The use of the Internet by women with breast cancer and men with prostate cancer—results of online research. *Journal of Public Health*, 16(2), 133-143.
- [27] Lambert, S. D., & Loiselle, C. G. (2007). Health information—seeking behavior. *Qualitative health research*, 17(8), 1006-1019.
- [28] Lee, Y. J., Boden-Albala, B., Larson, E., Wilcox, A., & Bakken, S. (2014). Online health information seeking behaviors of Hispanics in New York City: a community-based cross-sectional study. *Journal of medical Internet research*, 16(7), e176.
- [29] Leist, A. K. (2013). Social media use of older adults: a mini-review. *Gerontology*, 59(4), 378-384.
- [30] Liu, K., Chen, Y., Lin, R., & Han, K. (2020). Clinical features of COVID-19 in elderly patients: A comparison with young and middle-aged patients. *Journal of Infection*.
- [31] Lovari, A. (2020). Spreading (dis) trust: Covid-19 misinformation and government inter-

- vention in Italy. *Media and Communication*, 8(2), 458-461.
- [32] Manierre, M. J. (2015). Gaps in knowledge: tracking and explaining gender differences in health information seeking. *Social Science & Medicine*, 128, 151-158.
- [33] McCloud, R. F., Okechukwu, C. A., Sorensen, G., & Viswanath, K. (2016). Entertainment or health? Exploring the internet usage patterns of the urban poor: a secondary analysis of a randomized controlled trial. *Journal of medical internet research*, 18(3), e46.
- [34] Mehra, M. R., Desai, S. S., Kuy, S., Henry, T. D., & Patel, A. N. (2020). Cardiovascular disease, drug therapy, and mortality in COVID-19. *New England Journal of Medicine*.
- [35] Onder, G., Rezza, G., & Brusaferro, S. (2020). Case-fatality rate and characteristics of patients dying in relation to COVID-19 in Italy. *Jama*, 323(18), 1775-1776.
- [36] Rooks, R. N., Wiltshire, J. C., Elder, K., BeLue, R., & Gary, L. C. (2012). Health information seeking and use outside of the medical encounter: is it associated with race and ethnicity?. *Social Science & Medicine*, 74(2), 176-184.
- [37] Rovetta, A., & Bhagavathula, A. S. (2020). Covid-19-related web search behaviors and infodemic attitudes in italy: Infodemiological study. *JMIR public health and surveillance*, 6(2), e19374.
- [38] Rowley, J., Johnson, F., & Sbaffi, L. (2017). Gender as an influencer of online health information-seeking and evaluation behavior. *Journal of the Association for Information Science and Technology*, 68(1), 36-47.
- [39] Sapienza, A., & Falcone, R. (2016). A Bayesian Computational Model for Trust on Information Sources. In *WOA* (pp. 50-55).
- [40] Schultz, F., Utz, S., & Göritz, A. (2011). Is the medium the message? Perceptions of and reactions to crisis communication via twitter, blogs and traditional media. *Public relations review*, 37(1), 20-27.
- [41] Siegrist, M., & Zingg, A. (2014). The role of public trust during pandemics: Implications for crisis communication. *European Psychologist*, 19(1), 23-32. <https://doi.org/10.1027/1016-9040/a000169>
- [42] Who, D. G. (2011). Report of the review committee on the functioning of the international health regulations (2005) in relation to pandemic (H1N1) 2009. Sixty-fourth World Health Assembly: World Health Organization, 49-50.
- [43] Wilensky, U. (1999). Center for connected learning and computer-based modeling. In *NetLogo*. Northwestern University.
- [44] Zucco, R., Lavano, F., Anfosso, R., Bianco, A., Pileggi, C., & Pavia, M. (2018). Internet and social media use for antibiotic-related information seeking: Findings from a survey among adult population in Italy. *International journal of medical informatics*, 111, 131-139.