Intelligent Agents from Symbolic to Neurosymbolic Systems: The Quest for Integration

Andrea Agiollo[1(✉)], Roberta Calegari[2], Giovanni Ciatto[2], Matteo Magnini[2], Andrea Omicini[2], and Federico Sabbatini[3]

[1] Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), Delft University of Technology, Delft, The Netherlands
A.Agiollo-1@tudelft.nl
[2] Department of Computer Science and Engineering (DISI), Alma Mater Studiorum–Università di Bologna, Bologna, Italy
{roberta.calegari,giovanni.ciatto,matteo.magnini,andrea.omicini}@unibo.it
[3] Department of Pure and Applied Sciences (DiSPeA), Università degli Studi di Urbino Carlo Bo, Urbino, Italy
f.sabbatini1@campus.uniurb.it

Abstract. In this chapter we take as our reference twenty-five years of scientific and technical results presented at the Workshop on Objects, and explore the development of *rational agents* and integration with machine learning (ML) techniques, discussing their transition from pure *symbolic* to *subsymbolic* and *neurosymbolic* systems. Given the growing importance of combining rational agent reasoning with ML, we first outline the current state of technology by highlighting key milestones and breakthroughs. Pinpointing logics and logic programming as the main foundational tool for the design and implementation of rational agents, we discuss successful implementations and applications of *logic-based agents*, then we identify some of the main integration strands of subsymbolic techniques within rational agents. In particular, we focus on *symbolic knowledge injection (SKI)* and *symbolic knowledge extraction (SKE)* as some of the most relevant neurosymbolic techniques, and on their impact on intelligent agents and multi-agent systems (MASs). Current gaps and challenges in the integration of rational agents with ML are finally discussed along with future research directions.

Keywords: Rational Agents · Logic Programming · Multi-agent Systems · Symbolic-Subsymbolic Integration

1 Introduction

In the twenty-five years of the Workshop on Objects and Agents (WOA), agents and multi-agent systems (MASs) have obviously been at the core of the scientific and technical discussion: after a few years when the computational paradigm for

© The Author(s) 2026
V. Mascardi and A. Omicini (Eds.): *The Agents Journey*, LNCS 16395, pp. 320–339, 2026.
https://doi.org/10.1007/978-3-032-22940-3_12

distributed and situated systems was the main focus – as the " objects vs. agents" debate –, the general attention critically shifted towards intelligent agents and their role in the engineering of intelligent MASs. There, mainly, the notion of intelligent agent mostly referred to rational agent architectures, exploiting classical *symbolic* artificial intelligence (AI) approaches such automated reasoning, and exploiting the power of logics – e.g., in belief-desire-intention (BDI) architectures and frameworks [12] – and logic programming—for logic-based, or, *logic agents* [25].

In the last years, instead, the explosion of the general interest towards deep learning as one of the main trend in intelligent systems has led to the re-emergence of the so-called *AI agents* – a revamped term from the Nineties, see e.g. [29] –, nowadays understood as technical abstractions aimed at encapsulating machine learning (ML)-based *subsymbolic* techniques, so as to make their exploitation possible as legitimate interoperable components of intelligent systems. However, while on the one hand the strictly-cognitive properties of ML-based agents are up to some debate, their overall ability to exhibit sophisticated forms of intelligent behaviour is no longer disputable. Unlike *symbolic agents*, *subsymbolic agents* are already fast and effective, and behave well in real-world application scenarios, and capable of dealing efficiently with vast amounts of heterogeneous and possibly partially-specified or incomplete data. At the same time, any subsymbolic approach to AI inherently brings about the issues of transparency and interpretability of intelligent behaviours, mostly preventing actual human understanding of AI systems [15]—thus opening new critical problems, such as trustability and *explainability*, e.g. [2,77].

Rational agents – exploiting symbolic techniques based on first-order cognitive abstractions for their cognitive processes –, even with their well-known shortcomings in terms of computational efficiency and limited ability to deal with real-world scenarios, are instead to some extent *understandable by design*: their behaviour is typically easy to be interpreted by both intelligent agents and humans. For instance, a rational planning agent could straightforwardly justify and explain any action from its observable behaviour by just sharing its view of the world and its goals, along with its rational plan. Generally speaking, rational agents – such as logic agents, using formal logics as the foundation of their reasoning process –, offer a robust foundation for understanding and explaining decisions and behaviours by intelligent systems.

Thus, as AI systems become increasingly complex and ubiquitous, whereas the need for frameworks that are both interpretable and reliable has never been more critical [59], the integration of ML techniques within rational agents has clearly the potential of representing a transformative approach to AI, merging the strengths of symbolic reasoning with data-driven learning [25,89]. Blending symbolic reasoning with subsymbolic techniques, and integrating them in *neurosymbolic systems* – e.g., [84] – aims at addressing all of the aforementioned challenges by framing the real-world effectiveness of ML-based approaches within well-founded rational frameworks constraining and driving decision-making processes and intelligent behaviours [26].

Despite that promise of integration, however, several gaps and challenges still persist. Issues such as scalability, biases inherent in data, and the reliability of decision-making processes in high-stakes environments need to be critically examined. On the other hand, the roles that rational agents could potentially play within many different critical sectors – including healthcare, finance, autonomous systems, and smart cities – highlight the practical implications of this research line.

Accordingly, in this chapter we start by sketching the current landscape of rational agents, focussing on logic-based agents (*logic agents*, henceforth), as well as on their integration with subsymbolic techniques, typically as they emerge from the ML field. After discussing the existing challenges, we point out possible future directions for research, by exploiting the twenty-five years of WOA as our main reference.

There, rational and logic agents, along with the integration of symbolic and subsymbolic approaches in agents and MASs, have represented a recurring theme over the past twenty-five years. So, in the remainder of this chapter we first outline the key milestones and achievements in the field, then we introduce the two main integration strands that have characterised the most recent and effective developments: *symbolic knowledge extraction (SKE)* and *symbolic knowledge injection (SKI)*. We delve into these topics, by assessing their potential to enhance the interpretability and explainability of intelligent systems, and also by addressing specific relevant issues such as their ability to improve ethical alignment of AI systems to human values. Finally, we analyse current gaps, challenges, and future directions towards the effective integration of symbolic and subsymbolic techniques.

2 Logic and Rational Agents in Multi-agent Systems

The scientific and technical histories of computational logic and MASs are deeply intertwined. As remarkable examples, well-known workshop lines such as CLIMA (Computational Logic in Multi-Agent Systems)[1] and DALT (Declarative Agent Languages and Technologies)[2] have carried on and developed that specific scientific landscape for more than a decade. Overall, in the years, logic has proven to be a powerful tool for modelling, verifying, and reasoning about the behaviour of agents in complex systems [14].

Computational logic provides the foundation for many aspects of MASs—way beyond formal specification, as one may think at first glance. The history of WOA itself – which started by focussing on the power of objects and agents as abstractions for distributed systems, and nowadays is instead mostly devoted to elaborate on the role of multi-agent systems in the engineering of intelligent systems – corroborates such a connection, as demonstrated by the large number of contributions about logic that can be found in WOA proceedings. In fact, WOA has featured papers about the many different aspects of logic for several

[1] https://dblp.org/db/conf/clima/.
[2] https://dblp.org/db/conf/dalt/.

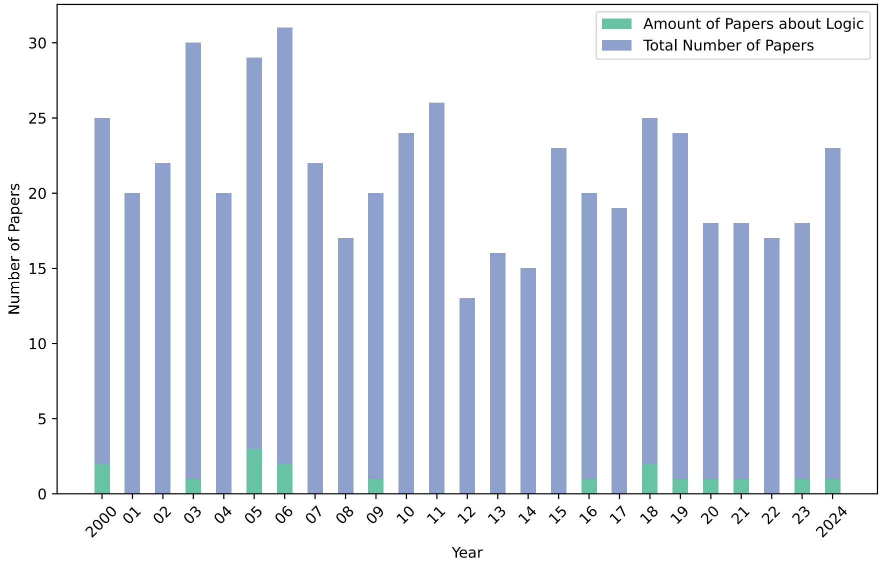

Fig. 1. Number of papers on logic-based approaches in MASs at WOA per year (2000–2024).

years since the inception of the workshop in 2000—as shown in Fig. 1. The recurring, almost continuous presence of those sorts of contributions straightforwardly emphasises the enduring relevance of logic in the area of intelligent agent and MAS research.

More in detail, logic has played a range of different roles in the MAS area, serving as a cornerstone for declarative specification, communication protocols, formal verification, agent programming, and – more recently – hybrid systems integrating ML techniques with MASs. Below, we proceed by categorising the many roles that logic has played in the MAS fields according to the papers that can be found in the twenty-five years of WOA proceedings.

2.1 Declarative Specification

Declarativeness is a warhorse of programming languages based on computational logic: by evaluating the truth value of logic formulas, logic-based programs can just specify *what* a machine should do, instead of *how* to do it [20]. In particular, logic programming languages like Prolog are amenable of both a declarative and an operational interpretation, and can exploit the fact that the two corresponding semantics match [58], so that they can be used as languages for executable specification. This makes logic-based systems particularly suitable for MASs when it comes to create high-level agent specifications or programs, as well as when agents have to reason about their environment, or interact with other agents.

Contributions focussing on declarativeness are manifold. For instance, in [13] a declarative language aimed at customising agent-based systems is discussed, focussing on interaction with web services and interoperability. There, Baldoni *et al.* highlight how personalised service selection and composition – particularly in web-based and multi-agent system – can benefit from reasoning about actions and conformance to policies—which are made simple by declarative programming.

Similarly, Costantini *et al.* [43] propose a novel method for expressing preferences within logic-based agent languages. There, agents can define preferences in the body of rules, allowing them to choose among multiple actions based on evolving preferences, via a declarative language. The paper discusses how that approach makes it possible to support dynamic decision-making in agents, enabling them to modify and adjust their actions as their goals and priorities change over time.

Declarativeness is also a key concern in [34], where Ciatto *et al.* discuss the interplay between blockchain technology and logic programming, and argue that logic programming has the potential to enhance the intrinsic capabilities of blockchain—in particular in the development of smart contracts. There, the authors argue that logic-based approaches could make smart contracts more expressive and reliable, whereas blockchain could in turn provide logic-based systems with the distributed infrastructure they typically need in real-world scenarios.

Finally, a timed epistemic logic is introduced in [44], aimed at modelling cooperation and belief changes within groups of agents. There, logic is used to formalise agents willing to either join or leave groups, and even agents temporarily "lent" to other groups. By leveraging on temporal reasoning, the system is able to express and check the validity of agents' beliefs and knowledge, making it particularly useful for dynamic, time-sensitive MASs.

2.2 Communication and Interoperability

In MASs, agents often need to communicate and collaborate across distributed systems. To this end, logic has been mostly exploited so as to ensure that agents can interact with each other effectively, while adopting specific communication protocols, as well as ensuring system interoperability. For instance, this is the case of the aforementioned work by Baldoni*et al.* [13], as well as of the work by Chesani *et al.* [32], which is discussed later in this section. Another example is [33], where Ciampolini *et al.* propose a language for coordinating abductive logic agents, focusing on how agents can interact within MASs, either they are competing or collaborating. The language allows agents to solve problems based on incomplete knowledge by formulating hypotheses via adductive reasoning. In order to showcase the ability of the proposed system to handle both collaboration and competition among agents, the paper presents a medical diagnosis example.

Furthermore, computational logic has also been exploited as the foundation for "programming the interaction space" [38] within MASs, exploiting a declarative model for the coordination [37] of both intelligent and non-intelligent agents – namely, TuCSoN [71] – aimed at enabling and governing both the explicit communication and the stigmergic interaction of agents within MASs. Along this line, WOA proceedings are disseminated with contributions, which we mention in chronological order on the following.

First of all, Ricci *et al.* [76] demonstrate how the TuCSoN coordination infrastructure helps manage autonomous agents in network-centric applications using intelligent light management as a case study. The notion of *agent coordination contexts* (ACC) in TuCSoN is introduced in [75], enabling agents to model, interact, and affect their environment while providing a flexible framework for MAS organisation. Omicini *et al.* [68] investigate the combination of *subjective* (intra-agent) and *objective* (inter-agent) *coordination* within FIPA agents using the TuCSoN framework. Oliva *et al.* [66,67] focus on how the TuCSoN coordination model and technology support multi-agent-based simulation (MABS), by showing how to simulate Minority Game with TuCSoN, with the aim of studying emergent behaviour in complex systems using logic-based agents for flexible and controllable simulations. Nardini *et al.* [65] explore a chemistry-inspired approach for the coordination of services in pervasive computing environments, where the ReSpecT logic-based coordination engine is used to enable adaptive and self-organising behaviour of TuCSoN agents. The main guidelines for a methodology specifically targeting the issue of *situatedness* in MASs are introduced in [63]; based on the TuCSoN coordination model, the methodology is meant to be centred around the notion of coordination as the key to managing dependencies between agents and their environment. Finally, Mariani *et al.* [64] try and go beyond logic towards other declarative mainstream languages for MAS coordination, by integrating XPath, blockchain, and stream processing with tuple-based coordination models so as to enhance MAS expressive power in the Internet of Things (IoT) as well as in general pervasive intelligence scenarios.

The generalised exploitation of logic programming engines at the core of distributed architectures – obviously including distributed MASs – has also been the subject of exploration in the early works by Calegari *et al.* [24], leading to the notion of logic programming as a service (LPaaS, which first appeared in the WOA proceedings in 2016. For instance, in [27] the authors introduce tuProlog as a lightweight logic programming engine suitable to provide rational agents with *as-a-service* logic programming capabilities – such as interactive goal demonstration – in distributed systems. Later on, Calegari *et al.* [23] explore the potential of computational logic in the realm of spatial and temporal reasoning by extending the LPaaS model and architecture towards *situated intelligence*, allowing agents to improve on their ability to take context-aware, location-specific decisions.

2.3 Formal Verification and Semantics

As in many other sorts of computational systems, logic has been instrumental in verifying the behaviour of agents within MASs, typically by ensuring that their interactive behaviour conforms to expected communication protocols and rules. This is especially crucial when engineers are concerned with the reliability of distributed, open, and dynamic MASs.

For instance, verification – see also [4] in this book – plays a primary role in [13] and [32], where the results of the MASSIVE projects are framed and shared. In particular, a logic-based framework for the specification and verification of agent interaction protocols in distributed MASs is discussed; examples in applications scenarios – e-commerce, medicine, and e-learning – are discussed. The framework leverages on formal logic for protocol specification, so that the compliance of agent observable behaviour with the protocols can be automatically verified, thus ensuring agent and MAS correctness.

Armando *et al.* [5] push formal verification even further by introducing the Logic Broker Architecture aimed at facilitating the integration of automated verification systems, such as theorem provers and model checkers, which commonly operate in isolation. The architecture enables different verification systems to interoperate by means of a registration/subscription mechanism and a translation process that ensure the safe and sound exchange of logical services across different systems.

2.4 Agent Programming, Reasoning and Planning

Another topic that is recurrent in the twenty-five years of WOA is obviously agent-oriented computing (AOC), and more specifically *agent programming*. Whenever rational agents are involved, agent programming mostly deals with agent *reasoning* and *planning*, where logics obviously play a crucial role. Rational agent architectures are generally designed around specific logics – e.g., BDI logic [74] –, so as to enable the declarative specification of actions, goals, beliefs, and plans, while ensuring sound agent decision making, deliberation, and planning processes.

One remarkable example of logic-based agent language is DALI, which firstly appeared in the WOA proceedings in 2003 [45]. There, Costantini *et al.* demonstrate how DALI can be used to implement STRIPS-like planning, allowing agents to plan and execute actions to achieve their goals. The DALI framework enables agents to engage in proactive behaviour by continuously checking for goals and actions, making it suitable for dynamic and adaptive environments.

Another example is the work by Magnolo *et al.* [62], which leverages fuzzy logic and ontologies so as to simulate crowd behaviour at large social events such as concerts. By analysing factors like crowd density and duration, the model aims at providing for more accurate simulations of crowd dynamics, which can be used to inform the design of safer and more efficient event management strategies.

More recently, the focus shifted towards the exploitation of the Prolog language [57] for agent programming. For instance, Ciatto *et al.* [36] introduce a

Kotlin-based domain-specific language (DSL) for Prolog, embedding logic programming directly into Kotlin – via the 2P-Kt ecosystem [35] –, thus providing a seamless way for object-oriented and functional developers to leverage logic-based reasoning within their projects—which could be expected to help promoting wider adoption of logic programming in mainstream programming environments. As expected, later on, the same approach propelled the development of [12], a BDI framework based on Kotlin.

Finally, Bordini *et al.* [19] compare agent-oriented logic programming languages in the literature, showing how those languages extend traditional logic programming (most of them leveraging on Prolog) so as to model autonomous agents and MASs. There, the practical applications of those languages in the context of complex, real-world MASs are discussed, highlighting their utility in scenarios requiring autonomous decision-making.

3 Rational Agents and Machine Learning in Multi-agent Systems

3.1 Hybrid Systems

More recent developments in MAS research try and bridge agents and ML, to let agents learn either from data or from other agents. In all cases, logic lies in the middle, enabling the definition of hybrid systems that leverage both symbolic and/or subsymbolic approaches.

For instance, the work by Costantini *et al.* [46] introduces a method for agents to learn through the exchange of rule sets, as a form of *cultural transmission* akin to human learning: in this case, learning is purely symbolic. Agents evaluate new knowledge based on its potential for the achievements of specific goals, and are equipped with mechanisms for discarding unhelpful or incorrect information. Overall, this allows agents in a MAS to adapt and improve their behaviour through cooperative knowledge sharing.

Conversely, D'Asaro *et al.* [47] introduce a system that combines subsymbolic ML and logic-based techniques to assist in motor rehabilitation. The system uses data from multiple sources (e.g., sensors) in order to model users' cognitive and motor abilities, and help therapists making decisions in the evaluation and adjustment of rehabilitation exercises. There, the intrinsic transparency of logic-based approaches allows for an easier understanding of decisions by human patients, which is critical in therapeutic settings.

Many other works that explore the integration of logic and machine learnin in multi-agent system can be found in the WOA proceedings, especially in recent years, when the field has seen a surge in hybrid systems that aim at combining the strengths of both approaches. Accordingly, here and in the next subsections, we explore the related contributions, highlighting some of the most relevant slices of literature.

3.2 Towards Neurosymbolic Agents

The integration of logic agents and ML is transforming the AI landscape by merging the strengths of symbolic reasoning with the adaptability, effectiveness, and efficiency of data-driven approaches (subsymbolic techniques). At the core of the integration is typically the ability of logic agents to represent complex knowledge using formal logic, which provides them with a structured foundation for their reasoning process. Meanwhile, ML excels in learning patterns and making predictions from large datasets, allowing systems to adapt and improve based on empirical evidence. Such a symbiotic relationship enhances the capabilities of intelligent systems across various domains.

One significant area where synergies are likely to be found is knowledge representation and learning, where logic programming can be used to define a rich structure for a problem domain, providing for a formal framework for representing rules, relationships, and constraints. A structured representation in principle enables ML to continuously refine the knowledge base through data-driven insights, thus enabling or facilitating a dynamic learning environment. As ML algorithms analyse incoming data, they can update and adjust the logic framework, thus ensuring that the system not only adheres to logic constraints, but also evolves based on new information—and its evolution fosters greater accuracy and reliability in decision-making processes.

Furthermore, the inclusion of logical constraints into ML models helps harnessing the learning processes by embedding domain-specific knowledge within the model. Techniques such as constraint-based learning allow for the specification of acceptable solution spaces, ensuring that the learned models stay consistent with respect to the established domain knowledge. This approach can improve the robustness of the decision-making process by preventing the model from exploring infeasible or irrelevant solutions. Other techniques, such as neurosymbolic approaches, further blend neural networks with symbolic reasoning, creating systems that ideally are able to leverage on the strengths of both paradigms. By combining the generalisation capabilities of neural networks with the rationality of symbolic reasoning, integrated neurosymbolic systems can achieve a more nuanced understanding of complex tasks.

Finally, and most interestingly, recent advancements have introduced promising techniques for SKI and SKE, which further enhance the integration of LP and ML. Symbolic extraction focuses on the derivation of interpretable models from complex ML systems, allowing for the extraction of meaningful patterns and rules. Conversely, symbolic injection involves incorporating logical rules into ML frameworks, thus enriching the learning process with structured knowledge. Those techniques have exhibited significant potential in improving the transparency and reliability of AI systems, and many recent works presented at WOA over the years have explored their applications in depth—so, they will be detailed further in the remainder of this section.

3.3 Symbolic Knowledge Extraction in Intelligent Agent Systems

When transposing the ML workflow into the realm of logic agents, symbolic knowledge can be seen as a common means to enable agent communication, information sharing, and cooperation. Symbolic knowledge is produced in several ways—e.g., it may be encoded by domain experts, or, distilled via dedicated SKE algorithms. More in detail, SKE focuses on extracting a symbolic representation out of the knowledge acquired by subsymbolic predictors. This approach offers a higher degree of interpretability for opaque ML predictions, making it possible to inspect the subsymbolic model. Here and in the following we stick to the definition provided in [39]: accordingly, SKE refers to

> any algorithmic procedure accepting trained subsymbolic predictors as input and producing symbolic knowledge as output, so that the extracted knowledge reflects the behaviour of the predictor with high fidelity

In the context of intelligent agents, SKE provides effective tools to inspect the agents' knowledge and behaviour.

Generally speaking, the main benefit of SKE for intelligent agents is that it makes it possible to inspect agents' knowledge—for instance, to check whether it satisfies domain requirements, fairness constraints, or other behavioural specifications. Even as a recent technique, SKE has already found practical application in a wide range of fields, including credit-risk evaluation [10,11,87], credit card screening [85], intrusion detection in computer networks [56], keyword extraction [7], space mission diagnostic [81], and in the medical field as well [18,53,55]. Overall, the apparent wide applicability of SKE techniques suggests that neurosymbolic agents based on SKE have the potential to extend MAS effective operation over a large range of diverse real-word scenarios.

The actual applicability of SKE techniques is enhanced by the availability of public software libraries implementing state-of-the-art extraction algorithms, such as PSyKE,[3] `ruleex`,[4], and `Rule_Extraction_From_Trees`.[5]

In particular, PSyKE [28,79] – first presented at WOA [78] – is a Python framework supporting a significant set of pedagogical SKE techniques while maintaining complete compatibility with other mainstream Python packages as Scikit-Learn [72], also supporting Semantic Web interoperability for intelligent agents [80].

3.4 Symbolic Knowledge Injection in Intelligent Agent Systems

Quite intuitively, SKI serves a dual purpose w.r.t. to SKE, since it constrains an ML predictor to take into account some symbolic knowledge when drawing predictions. According to the definition provided in [39], SKI refers to

[3] https://github.com/psykei/psyke-python.
[4] https://github.com/fantamat/ruleex.
[5] https://github.com/Yimeng-Zhang/Rule_Extraction_from_Trees.

any algorithmic procedure affecting how subsymbolic predictors draw their inferences in such a way that predictions are either computed as a function of, or made consistent with, some given symbolic knowledge

Overall, SKI accounts for a wide set of techniques aimed at improving subsymbolic systems – such as neural networks (NNs) – by "injecting" them with some sort of structured and explicit symbolic knowledge, available at the rational agent level, affecting their overall behaviour.

Depending on the agent's context and task, symbolic knowledge can represent different things such as the agent's beliefs, some well-known concepts, a societal norm, or, any desirable rule to be taken into account. Therefore, the agent can solve a task by exploiting both the symbolic knowledge – deductive process – and the subsymbolic system—inductive process. The goals of SKI can be manifold, including: *(i)* improving the agent's predictive performances—e.g., accuracy, precision, recall, etc., *(ii)* making the agent's decisions more robust—e.g., against data degradation, adversarial attacks, etc., *(iii)* making the agent's decisions more compliant with some normative or ethical constraints, *(iv)* reducing the amount of data needed to train the subsymbolic system, *(v)* reducing learning time by providing straight away the very knowledge that subsymbolic predictors would otherwise struggle to learn by processing huge amounts of data, *(vi)* preventing subsymbolic predictors from working as full black boxes during their training—hence avoiding the need for explanations.

SKI can be performed in different ways, depending on the desired outcome, properties, and the nature of the symbolic knowledge to be injected. According to the taxonomy proposed in [39], SKI can be classified into three main categories:

predictor structuring – where the structure of the subsymbolic predictor is extended in order to accommodate the symbolic knowledge. This can be done by introducing new items – new layers, new neurons, new connections, new activation functions – that are specifically designed to mimic the symbolic knowledge. Relevant examples of this approach are knowledge based artificial neural networks (KBANNs) [88], fibred neural networks (FNNs) [54], logic tensor networks (LTNs) [9], and knowledge injection via network structuring (KINS) [60].

guided learning – where the training process of the subsymbolic predictor is altered so as to take into account the symbolic knowledge. This is usually done by modifying the loss function—for instance, by adding a new regularisation term that penalises the predictor when it does not comply with the symbolic knowledge. Examples of this approach are guiding backpropagation by inserting rules (GBIR) [8], and knowledge injection via lambda layer (KILL) [61].

knowledge embedding – where the subsymbolic predictor is provided with the symbolic knowledge as additional input data. This can be done by converting the symbolic knowledge into some numerical form (a.k.a. embedding) that can be fed to the subsymbolic predictor [30,31,86].

In the context of intelligent agents, SKI methods can be exploited to pursue different aims, such as:

agent learning enrichment – where SKI lets *sub*-symbolic methods consume *symbolic* knowledge to either improve or enrich the agent's learning capabilities. Along this line, SKI improves ordinary ML tasks inside the agent workflow, by allowing the subsymbolic predictors to process (or, to take into account) the structured symbolic knowledge available at the agent level. The fundamental idea underlying those approaches is that there exist some concepts that can be either cumbersome or troublesome to learn from examples – e.g., syntactical concepts, semantics, etc. –: those concepts can represent complex beliefs that an agent has gained about the external environment, normative, or bounding expressions regarding the agent's acceptable actions—and much more. Therefore, the symbolic knowledge expressing those high-level concepts can be injected directly into the model to be used inside the agent workflow.

agent knowledge manipulation – where SKI enables the *sub*-symbolic manipulation of an agent's *symbolic* knowledge, by enabling subsymbolic predictors to handle it similarly to symbolic engines. In doing so, SKI supports classic symbolic AI tasks while aiming at boosting their performance, either extending their capabilities or improving their raw processing latency. In this context, common tasks are:

- **logic inference** in its many forms – e.g. deductive, inductive, probabilistic, etc. –, i.e. drawing conclusions out of a symbolic (KB);
- **information retrieval** looking for information in a symbolic KB;
- **KB completion** finding (and adding) missing information in a symbolic KB;
- **KB fusion** merging several KBs into a single one, taking care of (possibly, syntactically different) overlaps;

The key point here is supporting tasks where both inputs and outputs are symbolic in nature, but leveraging upon subsymbolic methods to gain speed, fuzziness, and robustness against noise [73].

Finally, in the context of SKI techniques, the usage of graph neural networks (GNNs) has recently gained popularity in tackling relevant tasks which are hard to formalise or solve in the logic realm, because of either their numerical nature or their algorithmic infeasibility [1].

4 Conclusion

In this chapter we explore the emergence of subsymbolic techniques in the area of multi-agent systems, and their integration with rational architectures for intelligent agents based on symbolic approaches, focussing in particular on the incorporation of machine learning techniques within logic-based agents.

By taking as our reference twenty-five years of scientific and technical results presented at the Workshop on Objects and Agents – from 2000 to 2024 –, we first overview the main reasons for combining rational agent reasoning with subsymbolic ML techniques, outlining the current state of models and technologies in the field. Special attention is given to computational logic in general, and logic programming in particular, as a foundational tool for designing rational agents: successful implementations and applications of logic-based agents are discussed, and the benefits derived from logic programming are emphasised. Some of the most prominent roles that logic has played in the MAS area in the last decades are pointed out—such as declarative specification, communication and interoperability, formal verification and semantics, agent programming, reasoning, and planning.

We then identify some of the main strands of integration of subsymbolic techniques within rational agents, mostly focussing on how logic-based agents can leverage on machine learning techniques, and showcasing some relevant cases where this sort of integration has actually been proven as effective. Some of the current gaps and challenges in the integration of rational agents with ML are discussed along with potential and promising future research directions. In particular, we finally focus on symbolic knowledge injection and symbolic knowledge extraction techniques: we summarise the key findings, and emphasise their remarkable potential in improving the transparency and reliability of intelligent MASs and, in general, of AI systems.

Acknowledgments. This chapter was partially supported by *(i)* the "ENGINES – ENGineering INtElligent Systems around intelligent agent technologies" project, funded by the European Union – Next Generation EU within the framework of the National Recovery and Resilience Plan NRRP – Mission 4 "Education and Research" – Component 2 – Investment 1.1 "National Research Program and Projects of Significant National Interest Fund (PRIN)" – Call PRIN 2022 – D.D. n. 104 of 02/02/2022, under grant number 20229ZXBZM, *(ii)* the "AEQUITAS" project funded by European Union's Horizon Europe research and innovation programme under grant number 101070363, *(iii)* PNRR – M4C2 – Investimento 1.3, Partenariato Esteso PE00000013 – "FAIR—Future Artificial Intelligence Research" – Spoke 8 "Pervasive AI' ', funded by the European Commission under the Next Generation EU programme.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Agiollo, A., Ciatto, G., Omicini, A.: Graph neural networks as the copula mundi between logic and machine learning: a roadmap. In: Calegari et al. [22], pp. 98–115. http://ceur-ws.org/Vol-2963/paper18.pdf
2. Agiollo, A., Siebert, L.C., Murukannaiah, P.K., Omicini, A.: From large language models to small logic programs: building global explanations from disagreeing local post-hoc explainers. Auton. Agent. Multi-Agent Syst. **38**, 1–33 (2024). https://doi.org/10.1007/s10458-024-09663-8
3. Alderighi, M., Baldoni, M., Baroglio, C., Micalizio, R., Tedeschi, S. (eds.): WOA 2024 – 25th workshop "From Objects to Agents. In: CEUR Workshop Proceedings, vol. 3735. Bard, AO, Italy (2024). http://ceur-ws.org/Vol-3735/

4. Ancona, D., Briola, D., Ferrando, A., Martelli, M., Mascardi, V.: 25 years of declarative agent technologies in Italy. In: Mascardi, V., Omicini, A. (eds.) The Agents Journey: Twenty-five Years of Multi-agent Systems at WOA. Springer, Lecture Notes in Computer Science - State-of-the-Art Surveys (2026)
5. Armando, A., Zini, D.: Towards interoperable mechanized reasoning systems: the logic broker architecture. In: Corradi et al. [40], pp. 70–75. http://lia.disi.unibo.it/books/woa00/pdf/14.pdf
6. Armano, G., Paoli, F.D., Omicini, A., Vargiu, E. (eds.): WOA 2003 – 4th workshop "From Objects to Agents". Pitagora Editrice Bologna, Villasimius, CA, Italy (2003). http://giuseppevizzari.github.io/WOA-proceedings-archive/woa-2003.html
7. Azcarraga, A., Liu, M.D., Setiono, R.: Keyword extraction using backpropagation neural networks and rule extraction. In: The 2012 International Joint Conference on Neural Networks (IJCNN 2012), pp. 1–7. IEEE (2012). https://doi.org/10.1109/IJCNN.2012.6252618
8. Bader, S., Hölldobler, S., Marques, N.C.: Guiding backprop by inserting rules. In: Garcez, A.S.D., Hitzler, P. (eds.) 4th International Workshop on Neural-Symbolic Learning and Reasoning (NeSy'08), CEUR Workshop Proceedings, vol. 366, pp. 19–22. CEUR-WS.org, Patras, Greece (2008). http://ceur-ws.org/Vol-366/paper-5.pdf
9. Badreddine, S., d'Avila Garcez, A.S., Serafini, L., Spranger, M.: Logic tensor networks. Artif. Intell. **303**, 103649 (2022). https://doi.org/10.1016/J.ARTINT.2021.103649
10. Baesens, B., Setiono, R., De Lille, V., Viaene, S., Vanthienen, J.: Building credit-risk evaluation expert systems using neural network rule extraction and decision tables. In: Storey, V.C., Sarkar, S., DeGross, J.I. (eds.) ICIS 2001 Proceedings, pp. 159–168. Association for Information Systems (2001). http://aisel.aisnet.org/icis2001/20
11. Baesens, B., Setiono, R., Mues, C., Vanthienen, J.: Using neural network rule extraction and decision tables for credit-risk evaluation. Manage. Sci. **49**(3), 312–329 (2003). https://doi.org/10.1287/mnsc.49.3.312.12739
12. Baiardi, M., Burattini, S., Ciatto, G., Pianini, D.: JaKtA: BDI agent-oriented programming in pure Kotlin. In: Malvone, V., Murano, A. (eds.) 20th European Conference on Multi-Agent Systems (EUMAS 2023), Lecture Notes in Computer Science, pp. 49–65. Springer, Napoli, Italy (2023). https://doi.org/10.1007/978-3-031-43264-4_4
13. Baldoni, M., et al.: Personalization, verification and conformance for logic-based communicating agents. In: Corradini et al. [41], pp. 177–183. http://lia.deis.unibo.it/books/woa2005/papers/25.pdf
14. Baldoni, M., Baroglio, C., Mascardi, V., Omicini, A., Torroni, P.: Agents, multi-agent systems and declarative programming: who, what, when, where, why, how? In: Dovier, A., Pontelli, E. (eds.) A 25 Year Perspective on Logic Programming. Achievements of the Italian Association for Logic Programming, GULP, LNAI: State-of-the-Art Survey, vol. 6125, chap. 10, pp. 200–225. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14309-0_10
15. Belle, V., Papantonis, I.: Principles and practice of explainable machine learning. Front. Big Data **4**(688969), 1–25 (2021). https://doi.org/10.3389/fdata.2021.688969
16. Bergenti, F. (ed.): WOA 2009 – 10th workshop "From Objects to Agents". Seneca Edizioni Torino, Parma, Italy (2009). http://www.ailab.unipr.it/woa09/papers/

17. Bergenti, F., Monica, S. (eds.): WOA 2019 – 20th workshop "From Objects to Agents", CEUR Workshop Proceedings, vol. 2404. Parma, Italy (2019). http://ceur-ws.org/Vol-2404/
18. Bologna, G., Pellegrini, C.: Three medical examples in neural network rule extraction. Physica Medica **13**, 183–187 (1997). https://archive-ouverte.unige.ch/unige:121360
19. Bordini, R.H., Costantini, S., Monaldini, A., Vozna, A.: From pure prolog to logic agent-oriented programming languages. In: Alderighi et al. [3], pp. 271–285. http://ceur-ws.org/Vol-3735/paper_20.pdf
20. Calegari, R., Ciatto, G., Denti, E., Omicini, A.: Logic-based technologies for intelligent systems: state of the art and perspectives. Information **11**(3), 1–29 (2020). https://doi.org/10.3390/info11030167
21. Calegari, R., Ciatto, G., Denti, E., Omicini, A., Sartor, G. (eds.): WOA 2020 – 21st workshop "From Objects to Agents". In: CEUR Workshop Proceedings, vol. 2706. Bologna, Italy (2020). http://ceur-ws.org/Vol-2706/
22. Calegari, R., Ciatto, G., Denti, E., Omicini, A., Sartor, G. (eds.): WOA 2021 – 22nd workshop "From Objects to Agents. In: CEUR Workshop Proceedings, vol. 2963. Bologna, Italy (2021). http://ceur-ws.org/Vol-2963/
23. Calegari, R., Ciatto, G., Mariani, S., Denti, E., Omicini, A.: Logic programming in space-time: the case of situatedness in LPaaS. In: Cossentino et al. [42], pp. 63–68. http://ceur-ws.org/Vol-2215/paper_11.pdf
24. Calegari, R., Ciatto, G., Mariani, S., Denti, E., Omicini, A.: LPaaS as micro-intelligence: enhancing IoT with symbolic reasoning. Big Data Cogn. Comput. **2**(3) (2018). https://doi.org/10.3390/bdcc2030023
25. Calegari, R., Ciatto, G., Mascardi, V., Omicini, A.: Logic-based technologies for multi-agent systems: a systematic literature review. Auton. Agents Multi-Agent Syst. **35**(1) (2021). https://doi.org/10.1007/s10458-020-09478-3
26. Calegari, R., Ciatto, G., Omicini, A.: On the integration of symbolic and sub-symbolic techniques for XAI: a survey. Intelligenza Artificiale **14**(1), 7–32 (2020). https://doi.org/10.3233/IA-190036
27. Calegari, R., Denti, E., Mariani, S., Omicini, A.: Towards logic programming as a service: Experiments in tu prolog. In: Santoro et al. [83], pp. 79–84. http://ceur-ws.org/Vol-1664/w14.pdf
28. Calegari, R., Federico, S.: The PSyKE technology for trustworthy artificial intelligence. In: Dovier, A., Montanari, A., Orlandini, A. (eds.) AIxIA 2022 – Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 13796, pp. 3–16. Springer, Udine, Italy (2023). https://doi.org/10.1007/978-3-031-27181-6_1
29. Castelfranchi, C.: Modelling social action for AI agents. Artif. Intell. **103**(1), 157–182 (1998). https://doi.org/10.1016/S0004-3702(98)00056-3
30. Chang, K., Yih, W., Yang, B., Meek, C.: Typed tensor decomposition of knowledge bases for relation extraction. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 1568–1579 (2014). https://doi.org/10.3115/v1/d14-1165
31. Chang, M., Ullman, T., Torralba, A., Tenenbaum, J.B.: A compositional object-based approach to learning physical dynamics. In: Proceedings of the 5th International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017. OpenReview.net (2017). https://openreview.net/forum?id=Bkab5dqxe
32. Chesani, F., et al.: Protocol specification and verification by using computational logic. In: Corradini et al. [41], pp. 184–192. http://lia.deis.unibo.it/books/woa2005/papers/26.pdf

33. Ciampolini, A., Lamma, E., Mello, P., Torroni, P.: Expressing collaboration and competition among abductive logic agents. In: Corradi et al. [40], pp. 64–69. http://giuseppevizzari.github.io/WOA-proceedings-archive/pdfs/woa2000/WOA13.pdf
34. Ciatto, G., Calegari, R., Mariani, S., Denti, E., Omicini, A.: From the blockchain to logic programming and back: research perspectives. In: Cossentino et al. [42], pp. 69–74. http://ceur-ws.org/Vol-2215/paper_12.pdf
35. Ciatto, G., Calegari, R., Omicini, A.: 2P-Kt: a logic-based ecosystem for symbolic AI. SoftwareX **16**, 1–7 (2021). https://doi.org/10.1016/j.softx.2021.100817
36. Ciatto, G., Calegari, R., Siboni, E., Denti, E., Omicini, A.: 2P-Kt: logic programming with objects & functions in Kotlin. In: Calegari et al. [21], pp. 219–236. http://ceur-ws.org/Vol-2706/paper14.pdf
37. Ciatto, G., Di Marzo Serugendo, G., Louvel, M., Mariani, S., Omicini, A., Zambonelli, F.: Twenty years of coordination technologies: COORDINATION contribution to the state of art. J. Logical Algebraic Methods Program. **113**(100531), 1–25 (2020). https://doi.org/10.1016/j.jlamp.2020.100531
38. Ciatto, G., Mariani, S., Omicini, A.: ReSpecTX: programming interaction made easy. Comput. Sci. Inf. Syst. **15**(3), 655–682 (2018). https://doi.org/10.2298/CSIS180111031C
39. Ciatto, G., Sabbatini, F., Agiollo, A., Magnini, M., Omicini, A.: Symbolic knowledge extraction and injection with sub-symbolic predictors: a systematic literature review. ACM Comput. Surv. **56**(6), 1–35 (2024). https://doi.org/10.1145/3645103
40. Corradi, A., Omicini, A., Poggi, A. (eds.): WOA 2000 – 1st workshop "From Objects to Agents", Atti di Congressi, vol. 1195. Pitagora Editrice Bologna, Parma, Italy (2000). http://giuseppevizzari.github.io/WOA-proceedings-archive/woa-2000.html
41. Corradini, F., De Paoli, F., Merelli, E., Omicini, A. (eds.): WOA 2005 – 6th workshop "From Objects to Agents". Pitagora Editrice Bologna, Camerino, MC, Italy (2005). http://lia.deis.unibo.it/books/woa2005/atti.pdf
42. Cossentino, M., Sabatucci, L., Seidita, V. (eds.): WOA 2018 – 19th workshop "From Objects to Agents". In: CEUR Workshop Proceedings, vol. 2215. Palermo, Italy (2018). http://ceur-ws.org/Vol-2215/
43. Costantini, S., Dell'Acqua, P., Tocchio, A.: Expressing preferences declaratively in logic-based agent languages. In: De Paoli et al. [49], pp. 138–143 .http://ceur-ws.org/Vol-204/P05.pdf
44. Costantini, S., Formisano, A., Pitoni, V.: A timed epistemic logic for formalizing cooperation among groups of agents. In: Falcone et al. [51], pp. 151–166. http://ceur-ws.org/Vol-3579/paper11.pdf
45. Costantini, S., Tocchio, A.: Strips-like planning in the DALI logic programming language. In: Armano et al. [6], pp. 115–120. http://giuseppevizzari.github.io/WOA-proceedings-archive/pdfs/woa2003/05.pdf
46. Costantini, S., Tocchio, A.: Learning by knowledge exchange in logical agents. In: Corradini et al. [41], pp. 1–8. http://lia.deis.unibo.it/books/woa2005/papers/1.pdf
47. D'Asaro, F.A., Origlia, A., Rossi, S.: Towards a logic-based approach for multimodal fusion and decision making during motor rehabilitation sessions. In: Bergenti and Monica [17], pp. 8–13. http://ceur-ws.org/Vol-2404/paper02.pdf
48. De Meo, P., Postorino, M.N., Rosaci, D., Sarnè, G.M.L. (eds.): WOA 2017 – 18th workshop "From Objects to Agents". In: CEUR Workshop Proceedings, vol. 1867. Scilla, RC, Italy (2017). http://ceur-ws.org/Vol-1867/
49. De Paoli, F., Di Stefano, A., Omicini, A., Santoro, C. (eds.): WOA 2006 – 7th workshop "From Objects to Agents". In: CEUR Workshop Proceedings, vol. 204. Catania, Italy (2006). http://ceur-ws.org/Vol-204/

50. De Paoli, F., Manzoni, S., Poggi, A. (eds.): WOA 2002 – 3rd workshop "From Objects to Agents". Pitagora Editrice Bologna, Milano, Italy (2002). http://giuseppevizzari.github.io/WOA-proceedings-archive/woa-2002.html
51. Falcone, R., Castelfranchi, C., Sapienza, A., Cantucci, F. (eds.): WOA 2023 – 24th workshop "From Objects to Agents". In: CEUR Workshop Proceedings, vol. 3579. Roma, Italy (2023). http://ceur-ws.org/Vol-3579/
52. Ferrando, A., Mascardi, V. (eds.): WOA 2022 – 23rd workshop "From Objects to Agents". In: CEUR Workshop Proceedings, vol. 3261. Genova, Italy (2022). http://ceur-ws.org/Vol-3261/
53. Franco, L., Subirats, J.L., Molina, I., Alba, E., Jerez, J.M.: Early breast cancer prognosis prediction and rule extraction using a new constructive neural network algorithm. In: Computational and Ambient Intelligence (IWANN 2007). LNCS, vol. 4507, pp. 1004–1011. Springer (2007). https://doi.org/10.1007/978-3-540-73007-1_121
54. d'Avila Garcez, A.S., Gabbay, D.M.: Fibring neural networks. In: McGuinness, D.L., Ferguson, G. (eds.) 19th National Conference on Artificial Intelligence, 16th Conference on Innovative Applications of Artificial Intelligence, pp. 342–347. AAAI Press/The MIT Press, San Jose, CA, USA (2004). https://cdn.aaai.org/AAAI/2004/AAAI04-055.pdf
55. Hayashi, Y., Setiono, R., Yoshida, K.: A comparison between two neural network rule extraction techniques for the diagnosis of hepatobiliary disorders. Artif. Intell. Med. **20**(3), 205–216 (2000). https://doi.org/10.1016/s0933-3657(00)00064-6
56. Hofmann, A., Schmitz, C., Sick, B.: Rule extraction from neural networks for intrusion detection in computer networks. In: 2003 IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pp. 1259–1265. IEEE (2003). https://doi.org/10.1109/ICSMC.2003.1244584
57. Körner, P., et al.: Fifty years of Prolog and beyond. Theory Pract. Logic Program. **22**(6), 776–858 (2022). https://doi.org/10.1017/S1471068422000102
58. Kowalski, R.A.: Predicate logic as programming language. In: Information Processing 74 – Proceedings of the 1974 IFIP Congress, pp. 569–574. North-Holland Publishing Company (1974). http://dblp.uni-trier.de/rec/html/conf/ifip/Kowalski74
59. Laato, S., Tiainen, M., Najmul Islam, A., Mäntymäki, M.: How to explain AI systems to end users: a systematic literature review and research agenda. Internet Res. **32**(7), 1–31 (2022). https://doi.org/10.1108/INTR-08-2021-0600
60. Magnini, M., Ciatto, G., Omicini, A.: KINS: knowledge injection via network structuring. In: Calegari, R., Ciatto, G., Omicini, A. (eds.) CILC 2022 – Italian Conference on Computational Logic. CEUR Workshop Proceedings, vol. 3204, pp. 254–267. CEUR-WS (2022). http://ceur-ws.org/Vol-3204/paper_25.pdf
61. Magnini, M., Ciatto, G., Omicini, A.: A view to a KILL: knowledge injection via lambda layer. In: Ferrando and Mascardi [52], pp. 61–76. http://ceur-ws.org/Vol-3261/paper5.pdf
62. Magnolo, E., Manenti, L., Manzoni, S., Sartori, F.: Towards a MAS model for crowd simulation at pop-rock concerts exploiting ontologies and fuzzy logic. In: Bergenti [16], pp. 92–99. http://www.ailab.unipr.it/woa09/papers/Magnolo.pdf
63. Mariani, S., Omicini, A.: TuCSoN coordination for MAS situatedness: Towards a methodology. In: Santoro and Bergenti [82], pp. 48–57. http://ceur-ws.org/Vol-1260/paper11.pdf
64. Mariani, S., Omicini, A., Ciatto, G.: Novel opportunities for tuple-based coordination: XPath, the Blockchain, and stream processing. In: De Meo et al. [48], pp. 61–64. http://ceur-ws.org/Vol-1867/w11.pdf

65. Nardini, E., Viroli, M., Casadei, M., Omicini, A.: A self-organising infrastructure for chemical-semantic coordination: experiments in TuCSoN. In: Omicini and Viroli [70], pp. 117–125. http://CEUR-WS.org/Vol-621/paper17.pdf
66. Oliva, E., Viroli, M., Omicini, A.: Minority game: a logic-based approach in TuCSoN. In: De Paoli et al. [49], pp. 181–186. http://ceur-ws.org/Vol-204/P02.pdf
67. Oliva, E., Viroli, M., Omicini, A.: Simulation of minority game in TuCSoN. In: De Paoli et al. [49], pp. 6–9. http://ceur-ws.org/Vol-204/D03.pdf
68. Omicini, A., Ricci, A., Rimassa, G., Viroli, M.: Integrating objective & subjective coordination in FIPA: a roadmap to TuCSoN. In: Armano et al. [6], pp. 85–91. http://giuseppevizzari.github.io/WOA-proceedings-archive/pdfs/woa2003/07.pdf
69. Omicini, A., Viroli, M. (eds.): WOA 2001 – 2nd workshop "From Objects to Agents". Pitagora Editrice Bologna, Modena, Italy (2001). http://giuseppevizzari.github.io/WOA-proceedings-archive/woa-2001.html
70. Omicini, A., Viroli, M. (eds.): WOA 2010 – 11th workshop "From Objects to Agents". In: CEUR Workshop Proceedings, vol. 621. Rimini, Italy (2010). http://ceur-ws.org/Vol-621/
71. Omicini, A., Zambonelli, F.: Co-ordination of mobile information agents in TuCSoN. Internet Res. **8**(5), 400–413 (1998). https://doi.org/10.1108/10662249810241266
72. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. (JMLR) **12**, 2825–2830 (2011). https://dl.acm.org/doi/10.5555/1953048.2078195
73. Rafanelli, A., Magnini, M., Agiollo, A., Ciatto, G., Omicini, A.: An empirical study on the robustness of knowledge injection techniques against data degradation. In: Alderighi et al. [3] , pp. 20–32. http://ceur-ws.org/Vol-3735/paper_02.pdf
74. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In: Van de Velde, W., Perram, J.W. (eds.) Agents Breaking Away, Lecture Notes in Computer Science, vol. 1038, pp. 42–55. Springer (1996). https://doi.org/10.1007/BFb0031845
75. Ricci, A., Omicini, A.: Agent coordination contexts: experiments in TuCSoN. In: De Paoli et al. [50], pp. 14–21. http://giuseppevizzari.github.io/WOA-proceedings-archive/pdfs/woa2002/17.pdf
76. Ricci, A., Omicini, A., Denti, E.: Enlightened agents in TuCSoN. In: Omicini and Viroli [69], pp. 101–106. http://giuseppevizzari.github.io/WOA-proceedings-archive/pdfs/woa2001/pdf/21.pdf
77. Sabbatini, F., Calegari, R.: Unveiling opaque predictors via explainable clustering: the CReEPy algorithm. In: Boella, G., et al. (eds.) Proceedings of the 2nd Workshop on Bias, Ethical AI, Explainability and the role of Logic and Logic Programming co-located with the 22nd International Conference of the Italian Association for Artificial Intelligence (AI*IA 2023), Rome, Italy, 6 November 2023. CEUR Workshop Proceedings, vol. 3615, pp. 1–14. CEUR-WS.org (2023). https://ceur-ws.org/Vol-3615/paper1.pdf
78. Sabbatini, F., Ciatto, G., Calegari, R., Omicini, A.: On the design of PSyKE: a platform for symbolic knowledge extraction. In: Calegari et al. [22], pp. 29–48. http://ceur-ws.org/Vol-2963/paper14.pdf
79. Sabbatini, F., Ciatto, G., Calegari, R., Omicini, A.: Symbolic knowledge extraction from opaque ML predictors in PSyKE: platform design & experiments. Intelligenza Artificiale **16**(1), 27–48 (2022). https://doi.org/10.3233/IA-210120

80. Sabbatini, F., Ciatto, G., Omicini, A.: Semantic web-based interoperability for intelligent agents with PSyKE. In: Calvaresi, D., Najjar, A., Winikoff, M., Främling, K. (eds.) Explainable and Transparent AI and Multi-Agent Systems, Lecture Notes in Computer Science, vol. 13283, chap. 8, pp. 124–142. Springer (2022). https://doi.org/10.1007/978-3-031-15565-9_8
81. Sabbatini, F., Grimani, C., Calegari, R.: Bridging machine learning and diagnostics of the ESA LISA space mission with equation discovery via explainable artificial intelligence. Adv. Space Res. **74**(1), 505–517 (2024). https://doi.org/10.1016/j.asr.2024.04.041
82. Santoro, C., Bergenti, F. (eds.): WOA 2014 – 15th workshop "From Objects to Agents". In: CEUR Workshop Proceedings, vol. 1260. Catania, Italy (2014). http://ceur-ws.org/Vol-1260/
83. Santoro, C., Messina, F., De Benedetti, M. (eds.): WOA 2016 – 17th workshop "From Objects to Agents". In: CEUR Workshop Proceedings, vol. 1664. Catania, Italy (2016). http://ceur-ws.org/Vol-1664/
84. Sen, P., de Carvalho, B.W.S.R., Riegel, R., Gray, A.G.: Neuro-symbolic inductive logic programming with logical neural networks. In: 36th AAAI Conference on Artificial Intelligence, pp. 8212–8219. AAAI Press (2022). https://doi.org/10.1609/aaai.v36i8.20795
85. Setiono, R., Baesens, B., Mues, C.: Rule extraction from minimal neural networks for credit card screening. Int. J. Neural Syst. **21**(04), 265–276 (2011). https://doi.org/10.1142/S0129065711002821
86. Spillo, G., Musto, C., de Gemmis, M., Lops, P., Semeraro, G.: Knowledge-aware recommendations based on neuro-symbolic graph embeddings and first-order logical rules. In: Golbeck, J., et al. (eds.) RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, 18–23 September 2022, pp. 616–621. ACM (2022). https://doi.org/10.1145/3523227.3551484
87. Steiner, M.T.A., Steiner Neto, P.J., Soma, N.Y., Shimizu, T., Nievola, J.C.: Using neural network rule extraction for credit-risk evaluation. Int. J. Comput. Sci. Netw. Secur. **6**(5A), 6–16 (2006). http://paper.ijcsns.org/07_book/200605/200605A02.pdf
88. Towell, G.G., Shavlik, J.W., Noordewier, M.O.: Refinement of approximate domain theories by knowledge-based neural networks. In: Shrobe, H.E., Dietterich, T.G., Swartout, W.R. (eds.) Proceedings of the 8th National Conference on Artificial Intelligence. Boston, Massachusetts, USA, July 29–August 3 1990, vol. 2, pp. 861–866. AAAI Press/The MIT Press (1990). http://www.aaai.org/Library/AAAI/1990/aaai90-129.php
89. Zhang, W., Valencia, A., Chang, N.B.: Synergistic integration between machine learning and agent-based modeling: a multidisciplinary review. IEEE Trans. Neural Netw. Learn. Syst. **34**(5), 2170–2190 (2021). https://doi.org/10.1109/TNNLS.2021.3106777

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (http://creativecommons.org/licenses/by-nc-nd/4.0/), which permits any noncommercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if you modified the licensed material. You do not have permission under this license to share adapted material derived from this chapter or parts of it.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.