

Autonomy in the Age of Knowledge Graphs: Vision and Challenges

Jean-Paul Calbimonte ✉ 

University of Applied Sciences and Arts Western Switzerland HES-SO, Sierre, Switzerland

Andrei Ciortea ✉ 

Institute of Computer Science, University of St.Gallen, Switzerland

Timotheus Kampik ✉ 

Department of Computing Science, Umeå University, Sweden

SAP Signavio, Berlin, Germany

Simon Mayer ✉ 


Institute of Computer Science, University of St.Gallen, Switzerland

Terry R. Payne ✉ 

Department of Computer Science, University of Liverpool, United Kingdom

Valentina Tamma¹ ✉ 

Department of Computer Science, University of Liverpool, United Kingdom

Antoine Zimmermann ✉ 

Mines Saint-Etienne, Université Clermont Auvergne, INP Clermont Auvergne, CNRS, UMR 6158 LIMOS, France

Abstract

In this position paper, we propose that Knowledge Graphs (KGs) are one of the prime approaches to support the programming of autonomous software systems *at the knowledge level*. From this viewpoint, we survey how KGs can support different dimensions of autonomy in such systems: For example, the autonomy of systems with respect to their environment, or with respect to organisations; and we discuss related practical and research challenges. We emphasise that KGs need to be able to support systems of autonomous software agents that are themselves *highly heterogeneous*, which limits how these systems may use KGs. Furthermore, these heterogeneous software agents may populate *highly dynamic environments*, which im-

plies that they require adaptive KGs. The *scale of the envisioned systems* – possibly stretching to the size of the Internet – highlights the maintainability of the underlying KGs that need to contain large-scale knowledge, which requires that KGs are maintained jointly by humans and machines. Furthermore, autonomous agents require *procedural knowledge*, and KGs should hence be explored more towards the provisioning of such knowledge to augment autonomous behaviour. Finally, we highlight the importance of *modelling choices*, including with respect to the selected abstraction level when modelling and with respect to the provisioning of more expressive constraint languages.

2012 ACM Subject Classification Computing methodologies → Multi-agent systems; Computing methodologies → Intelligent agents; Computer systems organization → Self-organizing autonomic computing; Computing methodologies → Knowledge representation and reasoning; Information systems → Semantic web description languages

Keywords and phrases Knowledge graphs, Autonomous Systems

Digital Object Identifier 10.4230/TGDK.1.1.13

Category Vision

¹ Corresponding author



Funding This research has received funding from the Swiss National Science Foundation under grant No. 189474 (*HyperAgents*), grant No. 213369 (*StreamDKG*), from the European Union’s Horizon 2020 research and innovation program under grant No. 957218 (*IntelliIoT*), and from the Horizon Europe program under grant No. 101092908 (*SmartEdge*).

Acknowledgements We would like to thank Chiara Ghidini for the useful discussions on modelling processes using KGs to support process execution, mining and discovery.

Received 2023-09-18 **Accepted** 2023-11-17 **Published** 2023-12-19

Editors Aidan Hogan, Ian Horrocks, Andreas Hotho, and Lalana Kagal

Special Issue Trends in Graph Data and Knowledge

Author Contributions All authors participated in the planning and continuous refinements of this work, and hence contributed to the abstract, *Introduction* and *Conclusions* sections. JPC primarily contributed to Sections 3.5, 3.1, and 5. AC primarily contributed to Sections 3.1 and 3.4. TK primarily contributed to Sections 3.2, 3.3, 3.4, and 5. SM primarily contributed to Sections 3 and 5, as well as to the main tenet of the paper. TRP contributed to Sections 1, 3, and 4, and to the general arguments in the paper. VT primarily contributed to Section 2 and Section 4, as well as to the main tenet of the paper. AZ primarily contributed to Section 2, as well as to setting the general directions of the paper.

1 Introduction: Autonomy

Knowledge exists in different forms and for different purposes. Some of this knowledge may describe how the world or the universe works, and thus is *ontological* in nature: It describes what entities exist, how they can be categorised, and how they relate to each other. This knowledge, coupled with *propositional knowledge* (i.e., beliefs or assertions, which themselves may be true or false), can be used by computational entities (e.g., autonomous agents), to take informed decisions as to whether action is currently required. However, such knowledge does not typically describe what an agent needs to do in order to achieve a specific goal, and cannot be directly used by an agent to decide on what their next action or task should be. Conversely, knowledge may hence be *procedural*, in that it describes *how* an agent can achieve a task, either in isolation or through collaboration with other agents or services within its environment. This is analogous to, for example, the use of a user-manual that contains information about how customers can assemble their furniture by themselves, or the instructions for an espresso machine that allows the user to independently diagnose and rectify malfunctions. Knowledge may also be *normative* [20] (i.e., expressing notions such as prohibition, permission, obligation, or dispensation), which can be used to describe communities of agents and their inherent organisation (for example, when describing institutions [63]). This knowledge may be represented by, for example, a charter for a group that defines the mission of that group and the different roles that members may adopt, together with corresponding obligations that constrain their autonomy to actions that are in conformance with the group’s objectives; or the terms of use of a service such as an auction house that defines the rules that must be followed when bidding for goods without explicitly hard-coding this behaviour, thus allowing bidders to decide whether or not to comply with the rules – or to understand the sanctions that may be imposed if the rules are violated.

These different types of knowledge sources hold much of the information necessary for agents – natural or artificial – to facilitate their *autonomy*. To express such knowledge, the use of *Knowledge Graphs (KGs)* [38] has gained prominence across numerous research areas and application domains, due to its interconnected nature, expressivity, and the increasing variety of tools and techniques to support querying and reasoning at scale. KGs build upon a graph representation whereby objects (e.g., things, actions, tasks, affordances, methods, and capabilities) are represented as *nodes* and the relationships between them as *edges*. This type of representation, instantiated in the Resource

Description Framework (RDF) data model [22], has been used not only to represent *ontological* definitions and *propositional* definitions, but also *normative* knowledge [33], as we discuss in Section 4. As such, we claim that KGs are particularly useful in facilitating *autonomy*; KGs can be used to identify the specific vocabulary needed to represent autonomy and the types of reasoning tasks that are supported by the associated ontological models. This can be facilitated through the development of KG schemas that model autonomous behaviour, including the definition of entities (in terms of modules or fragments of hierarchical knowledge) that represent both static (e.g., *agent*, *object*, or *scene*) and dynamic (e.g., *task*, *goal*, or *action*) aspects of autonomy. The possibility to reuse and augment KGs promotes interoperability, which is further supported through the existence of highly interconnected and expressive schemata.

Sharing the goal of making data machine-readable and machine-understandable, KGs are well-aligned with the Semantic Web, which today consists of a plethora of knowledge sources that are interlinked and discoverable through a variety of different mechanisms. This provides more than simple access to static knowledge: Online services, data sources, and IoT devices that perform actions and perceive their environment all make up a rich and dynamic resource that is inherently human focused. This human focus is primarily due to the Web's hypermedia-driven interaction model that has already made much of the (human) Web content intuitively explorable and usable by humans without specialised training. The emergence of the Semantic Web has facilitated machine-oriented access and comprehension to much of this knowledge and, through the integration of KGs, it has recently become an increasingly rich knowledge base that can support autonomous behaviour by artificial agents, i.e., autonomous software systems, on the Web. Already today, the Web enables a variety of heterogeneous stakeholders to achieve their goals; whether they are reactive agents that are triggered to perform actions based on a set of preconditions, or humans who meet, transact, and interact. Furthermore, experiences by and interactions between these heterogeneous agents, through services (such as marketplaces and electronic banking), or social media (such as collaborative software development or data curation) can be interpreted as early signs of life of an emerging global social machine [78].

In this paper, we argue that autonomy and KGs are mutually supportive and beneficial, with the potential for both to gain from the resulting synergy. While many of our observations also hold for intelligent agents in general (including humans), our paper is focused on Web-based software agents. And we argue that there is a need to create systems that support the *programming of such artificial agents at the knowledge level*. Our central claim is that many of the underlying tenets of autonomy can be represented using KGs, autonomous agents may emerge that can transact across the Web – without the need for developers to encode specific protocols that the agents should use to interact with Web resources, specific information about other agents that they may encounter, or specific information about normative relationships. Likewise, by providing the procedural frameworks that permit agents to interact with services and institutions that they may encounter, the agents can potentially operate with a similar sense of autonomy that human users exhibit. Furthermore, since this Web-embedded knowledge is not read-only, autonomous Web Agents² may construct and expose new knowledge, and they may evolve knowledge that they find to be stale. We explore how any autonomous agent on the Web – human as well as artificial – could benefit from such knowledge, and how they may evolve this knowledge in turn. We examine how KGs provide a natural abstraction for representing the knowledge required by autonomous agents on an open, dynamic Web, and consider the current limitations of using such a formalism for autonomy. Our aim is not to provide an exhaustive treatment of KGs, but rather to provide the foundations for discussing how KGs have been used or could be used to *support different aspects*

² See <https://www.w3.org/community/webagents/>

13:4 Autonomy in the Age of Knowledge Graphs

of autonomy. This allows us to map the different characteristics of autonomy to the knowledge needed to support them and their representation, thereby highlighting the role that KGs play in current autonomous applications and pave the way for identifying challenges and open problems.

We first provide a brief introduction to KGs and the way they are represented. This is followed, in Section 3, by a discussion of how KGs can be used to support autonomy with respect to its individual dimensions as identified by Castelfranchi and Falcone [17]; we also discuss the evolution of KGs themselves in such open environments. While we argue that KGs are well suited to overcome many of the obstacles on the way to the programming of agents on the knowledge level, many hurdles still remain, as identified in Section 4. We therefore present the limitations of KGs with respect to supporting autonomy – at the moment and in principle – in Section 5, and we distill open questions from our discussion in Section 6.

2 Knowledge Graphs from an Autonomous Agents' Perspective

Autonomous software agents can be characterised as computational systems that acquire data from their environment (both sensory and from interactions with others), and reason about some course of action that may bring them closer to achieving their goals. These actions are then realised; this could be through the usage of some service, collaboration with other agents (through an appropriate communication mechanism), the creation of new knowledge, or the curation or modification of existing knowledge. This cycle repeats, typically without the direct intervention of humans in the goal-achievement process, which results in autonomous behaviour that can be proactive as well as reactive. However, for each step, the agents require a variety of different types of knowledge that may describe: their environment (see Section 3.1); other agents and their abilities (see Section 3.2); the relations, roles, and any form of normative organisation that they or others may be part of (see Section 3.3); and the structure of the problem domain (see Section 3.4). Although this knowledge is typically encoded implicitly or assumptions are made by many closed multi-agent systems, the ability to share this knowledge (e.g., for service discovery or to facilitate adherence to a normative institution) and to together evolve it (see Section 3.5) facilitates interoperability and reuse across different agent communities, which in turn can also increase reliability, veracity, and scalability for numerous agent tasks. By representing and sharing this knowledge in the form of KGs, we argue that the synergy and opportunities for agents within a knowledge- and service-rich environment are greatly increased, especially when the agents interact in open environments such as on the Web.

The use of KGs is today becoming more prominent across research areas and application domains, and within systems where graph-oriented data representation is used alongside relational databases or proprietary format files. Yet despite their increasing popularity, there is no single accepted definition of a KG [28]. In this context, we define a KG as a directed, edge-labelled multi-graph (whereby there may be different edges between two nodes that may have different relationships or types). In a KG, the *nodes* (or *vertices*) are labelled and represent real-world entities of interest (e.g., objects, actions, tasks, affordances, methods, and capabilities) whereas directed edges between a source and a target node are labelled with names that represents the relationships between these entities. This can lead to the simple formalisation of a KG as a set of triples “node–edge–node”, which is typically instantiated in the RDF data model [22]. However, other more involved structures may exist, such as property graphs [61] that allow nodes and edges to have a structure of their own. Whilst there is no commonly accepted definition of KGs, there is some agreement on the characteristics they should exhibit, based on the initial characterisation proposed by Paulheim [70], in that they:

1. represent (semantically-described) real-world entities and their interrelations by means of links, which are organised in a graph;
2. define entities and their relations by means of data models referred to as *ontologies* (vocabularies or schemata);
3. support potentially interrelating arbitrary entities with each other;
4. cover a variety of domains that span different topics and are often expressed at different levels of granularity, ranging from common-sense knowledge to domain- and task-specific knowledge.

KGs integrate both static data collections and stream data from multiple sources, thereby supporting sophisticated querying and exploration of this data in a seamless way [52]. This helps overcome the inherent abstraction gap that exists in the environment between raw sensor data and the higher-level knowledge needed by artefacts that produce and consume complex data to capture realistic models of the world. Thus, KGs have been used in a number of different contexts, including life sciences and linguistics, and a number of representations have been proposed. A symbolic language representation has typically been used by those KGs that aim to support autonomy, particularly those with precise semantics that support agreement on the vocabulary used and the reasoning services that infer new facts and check the consistency of the dataset. Semantic Web languages such as RDF³ and OWL [69] have become prominent candidates as: 1) they combine machine-readable syntax with a clear formal semantics; 2) they are sufficiently expressive to model vocabularies for disparate domains with sufficient detail; and 3) they provide extensive reasoning capabilities that range from type inheritance and type inference (RDFS) to richer reasoning services, e.g., classification, consistency checking, entailment checking, explanations and instance realisation (OWL), facilitated by the existence of different reasoner implementations.

Depending on the expressivity required by the downstream application, KGs can be distinguished as those that only possess relational statements in the form of RDF triples with no schema, and those that conform to an explicit schema (i.e., an *ontology*) that is expressed in RDFS or OWL. In the latter case, the expressivity of the schema depends on the constructors used in its axiom definitions.⁴ Expressive ontologies are highly interconnected, promote semantic interoperability between systems, and better support collaborative decision-making [1, 13]. However, despite efforts to promote the establishment of semantic mappings between entities [49, 66] and interlinking of data, data sources are often siloed, existing in isolation and are not interlinked with other sources, or they may be schema-less (or use a proprietary, unpublished, schema), with limited reuse or mappings to other data sources [36].

In addition to representing data and their schema in KGs using formal languages (i.e., RDF for data and RDFS or OWL for the schema), the Semantic Web stack also provides a standard for validating graphs in order to ensure the quality of data. The Shapes Constraint Language (SHACL) [46] is used to validate RDF graphs against a set of conditions described as “shapes” of data over nodes and edges on top of a KG schema expressed in OWL. Shapes are applied to “targets” and form a shapes graph defining constraints, i.e., the conditions against which a data graph is evaluated. A data graph is deemed valid if all the targets conform to their related shapes; furthermore, the result of the validation is a report expressed in RDF that can also be processed and queried using the same software used to manipulate the data graphs. SHACL has recently been used to model constraints of various type in virtual institutions and organisations [20]. The constraint components supported by SHACL are of two types: *core constraint* and *SPARQL-based constraint*. Core constraint components are those supported by SHACL-compliant systems, and can be divided into the following categories [68]:

³ RDF Syntax [77] and RDF Schema (RDFS) [15]

⁴ The direct semantics of OWL are expressed in Description Logic style and we refer the interested reader to Krötzsch et.al. [48] for a self-contained introduction to Description Logic.

13:6 Autonomy in the Age of Knowledge Graphs

- **Graph Structure Constraints:** These are constraints evaluated over the statements in the data graph and include, for example, minimum and maximum cardinality constraints satisfied by a focus node for some given paths, type of inference with respect to a focus node, and property pair equality and disjointness, verifying whether two sets of nodes reachable from different paths are equal or disjoint.
- **Filter Constraints:** These constraints are typically evaluated at node level, they are independent from the statements in the data graph and verify whether a focus node is of a particular type (e.g., IRI, literal or blank node); a given RDF term or a literal that satisfies certain conditions, e.g., a literal that is of type `xsd:integer`.
- **Logical Constraints:** These are standard logical operators (conjunction, disjunction, and negation) defined over other constraints.

SPARQL constraints support the ability to embed queries in SHACL constraints, therefore enhancing their expressivity. However, the inclusion of constraints that are potentially arbitrarily complex can come to the detriment of their usability and cause performance issues.

3 Knowledge Graphs and the Dimensions of Autonomy

Following our introduction of KGs from the perspective of autonomous agents, we next discuss in greater detail the different *types of knowledge* that support autonomous behaviour – in principle, and specifically on the Web. We structure this discussion according to the dimensions of autonomy that are introduced by Castelfranchi and Falcone [17] and concern the *environment* (see Section 3.1), *other agents* (see Section 3.2), *organisations* (see Section 3.3), and the *agent's* own goal-setting and execution (see Section 3.4). These are aligned with other meta-models for autonomous (multi-agent) systems, such as the JaCaMo meta-model [12] that distinguishes between the *environment*, *agent*, and *organisation* dimensions. We furthermore discuss the construction and evolution of knowledge *by* agents (see Section 3.5).

3.1 Knowledge supports Autonomy in Relation to the Environment

The first dimension of autonomy that we consider is the *autonomy of an agent in relation to its environment*. According to Castelfranchi and Falcone [17], this relates to agents being able to act in an environment based on their perception of this environment while maintaining autonomy from the perceived stimuli and from the environment itself. This is illustrated in [17] with a billiard ball – which is not autonomous with respect to its environment – and it is argued that agents who indeed are independent from their environment instead feature teleonomic behaviour, i.e., that they actively perceive and interpret their environment and the effects of their actions, that they orient themselves towards the input, and that their behaviour depends on internal states with their own evolution principles – rather than merely receiving an environmental “force” [17] as the billiard ball does. With respect to the *environment* dimension, the autonomy of an agent in an environment is increased when this agent gains knowledge about artifacts that are present (or likely present) in that environment. This includes knowing how it could successfully carry out *low-level interactions* with these artifacts – that is, information about the user interface or API of the artifact; it furthermore encompasses knowledge about how to *discover* artifacts in an environment (e.g., through an artifact registry or through broadcasting), and knowledge that permits an agent to establish that a specific artifact is *relevant* to an agent’s course of action – i.e., that using the artifact would be beneficial for the agent to achieve its goal.

The benefits of creating agents that are more autonomous from their environment are sought after across many fields: In *software engineering*, this relates to the creation of software systems that emphasise the non-functional properties of *adaptability* and *resilience* – that is, systems that

are still able to perform their core function even if their environment changes. These properties are typically – and, often, consciously – traded-off against *performance*: In a system that emphasises performance over adaptability, there is a high incentive to *tightly couple* even low-level interactions and to forgo run-time discovery capabilities by rather hard-coding device addresses and service APIs. Increasing the autonomy of a system with respect to its environment thus requires, explicitly or implicitly, to implement such systems in a way that does not couple their function to the *interfaces* of specific artifacts in the environment (e.g., by hard-coding a specific HTTP request); one layer above such interface coupling, the same goal motivates the creation of systems whose functions are not even coupled to specific *artifacts* – such systems hence need to be enabled to discover artifacts at run time [80], and then require an (internal or external) way to evaluate whether using these artifacts would contribute to the agents achieving their goals (e.g., a planner). The same fundamental principle is mirrored in *education*, a seemingly vastly different domain: In a similar way as it might be desirable to implement a software system that has enough knowledge to reason about the usage of specific artifacts while not being hard-coded to use them, we typically teach children in a way that increases their autonomy from *specific* environmental artifacts. To illustrate this, it is perfectly feasible to instruct a child – at the *interface* level – to use a specific means of public transport at a specific time and location to reach a specific destination – “To get to your Judo class, you need to always use bus line #2 at 5:37pm”. However, parents who emphasise autonomy of their child in relation to the environment will rather opt for explaining to the child – at the *knowledge* level – how a public transport system is used in general, which involves knowledge about public transportation, ticketing, routes, schedules, and possibly even how to read maps or use digital tools for navigation. This provides the child with information about how to navigate their environment effectively even when the environment is dynamic, and hence increases the child’s autonomy in the environment – e.g., when bus #2 is cancelled.

We argue that KGs are highly suited to support this type of autonomy of agents from their environment, i.e., with respect to the specifics of how to interact with things, by furnishing them with information about the artifacts in the environment and where this information is compatible with the information in the agents’ own knowledge base – in this case, this knowledge can be readily integrated and processed by the agent. Information about how to interact with (heterogeneous) artifacts in an environment would build on top of a basic (homogeneous) interaction ability; on the Web, this ability is hypermedia-based interaction, i.e., that an agent is able to use hypermedia controls (including unparameterised hyperlinks as well as forms). The first steps towards this have already been made, and have even been standardised by the World Wide Web Consortium (W3C): the goal of the *W3C Web of Things Thing Description*⁵ (W3C WoT TD) is precisely to permit agents that *understand* this standard to use any W3C WoT TD-described device they encounter in their environment. This is accomplished by conveying details about the usage of the interface of this device at run time. This is not only academically interesting, but has convincing practical application: For example, it means that a W3C WoT TD-described sensor can be upgraded to a better model without the requirement to update clients of the sensor, as long as the two sensors’ outputs are compatible with respect to the clients’ common knowledge. Further, while W3C WoT TDs today remain limited, they may be generalised to what is referred to as *Artifact Profiles*: “structured data describing the artifact through signifiers and general (domain- and application-specific) metadata” [87]. Such profiles hence describe features of an environmental artifact beyond its interface, and can be used by agents at run time to determine whether an artifact is interesting to them in their current situation – again increasing the agents’ environmental autonomy. It has also been proposed that such profiles could be brokered to agents

⁵ <https://www.w3.org/TR/wot-thing-description11/>

based on the current situation and abilities of the agent; this information is in this case shared in the form of an *Agent Profile* [54, 87]. In this scope, the SHACL vocabulary can be used to define behaviour specifications that describe to agents how to use an artifact (e.g., how to adjust a lamp’s brightness through an HTTP API) [83].

3.2 Knowledge supports Autonomy in Relation to other Agents

Regarding the second dimension of autonomy – *social autonomy* – the autonomy of an agent in its environment is also increased when this agent has sufficient knowledge about *other agents* that are present in this environment, which should not depend on the hard-wiring of agents’ social capabilities (either directly or through their environment). Similar to autonomy from the environment, we also consider social autonomy when training autonomous agents – this is less prevalent in software engineering, where software programs (themselves) are not yet being considered primarily as social entities, but it is highly visible in human education. In fact, this includes core tenets of education itself, including the ability to speak (multiple languages) and read, as well as neurological maturation. For example, the development of Theory of Mind (cf. [72]) in infants and, with it, the knowledge that one’s own beliefs, desires, emotions, and thoughts are different from those of others, is considered crucial for successful everyday social interactions among *heterogeneous* humans.

Following Castelfranchi and Falcone [17], we may distinguish between *autonomy as independence* – the ability of an agent to reach its goals without the help of other specific agents – and *autonomy in delegation*, i.e., an agent’s ability to achieve delegated tasks without detailed imperative instructions (colloquially: without *micro-management*). We argue that both of these sub-dimensions of autonomy can benefit from KGs in similar ways, as they require similar technical capabilities:

- i) self-sufficiency is a key requirement that provides the basic abilities necessary to achieve a goal or complete a delegated task, irrespective of the particular environment and the agents therein; and
- ii) in either case, generally applicable social communication abilities enable the utilisation of other agents’ abilities and resources in order to complete the goal or task *given* the current specifics of the environment.

In Section 3.1, we discussed how interface specifications can avoid the hard-wiring of two specific system interfaces to each other (including the tight coupling of agents to environment specifics), and introduced that this supports autonomy – specifically, by facilitating adaptability and resilience through dynamic (re)composability. In the same way, agents may avoid tight coupling *with other agents* by utilising knowledge about their social environment together with generic abilities to discover new knowledge by means of interacting with other agents. Concretely, and mirroring the environment dimension, such agents would not be hard-coded to interact with *specific* other agents in *specific* ways, but would rather be equipped with more broadly applicable knowledge about other agents – this knowledge might include that other agents might have specific abilities, that the agent might consult with them or delegate tasks to them, and that the likelihood of other agents to engage in an interaction is determined through social relationships between agents. An agent that is equipped with this knowledge – much like a human who understands social relationships – is much better suited to navigate a social situation, and hence features increased social autonomy.

KGs are very well suited to implement this type of knowledge-based programming, as they can augment an agent’s interactive capabilities and hence decrease its dependency on other *specific* agents. As in the case of autonomy from the environment, autonomous agents in this context may also benefit from both accessing external KGs and maintaining their own ones. Agents may then use internal KGs to maintain knowledge that is specific to the agent and knowledge that is

sufficiently important for being logically grouped with a given agent, thus facilitating this agent's independence. Some of the internal knowledge may be strictly private, while other knowledge can be shared with other agents, or be entirely public. For example, the agent may maintain lookup services of its own capabilities, used for provisioning and planning, many of which can be shared publicly, which facilitates collaboration with other agents. In contrast, knowledge about other specific agents that the agent maintains, e.g., for the purpose of strategic reasoning and negotiation, might remain private to start with. It may be disclosed eventually however, if by doing so, this is useful in the context of a specific objective, e.g., during a multi-agent dialogue. Finally, privacy-relevant knowledge such as health information about a natural person will typically remain private, but may be shared with authorised agents on a need-to-know basis. When KGs are applied within an agent, they can be seen as knowledge representation tools that are used to persist an agent's belief base, or snapshots thereof, considering that it may not be feasible to capture all belief revision operations that are executed in the course of the agent's reasoning cycle. In case external KGs are provided as part of the environment, agents might use them for *stigmery*, i.e., for coordination through environmental artefacts that are created or manipulated by the agents [73, 76] (see Section 3.1). This can facilitate multi-agent coordination and relax the coupling between agent instances, thus facilitating resilience and ability to respond to change.

3.3 Knowledge supports Autonomy in Relation to Organisations

Going beyond the interactions of agents where agent goals are generally aligned, *organisations* provide a deontic framework for agents to operate in by specifying joint objectives, norms, and policies. On the meta-level, they also permit agents to participate in organisational governance. The idea of virtual organisations as a means to facilitate autonomous (multi-agent) systems is a research direction that emerged at the turn of the century [18, 32, 67] and has since continued as a vibrant area of research [44]. While applications of multi-agent organisations in the sense of the holistic academic vision are nascent, most real-world organisations have by now some (often crucially important) form of electronic representation, and substantial efforts are incurred to ensure organisational compliance, e.g., with respect to privacy regulations [23].

In the context of organisational governance, KGs already exist in industry organisations for providing shared domain models that can be used across software systems and application scenarios, thus facilitating interoperability and centralising some of the knowledge maintenance efforts [40]. Beyond such current scenarios, at the organisational level, KGs are well-suited to enable agents to adjust to and switch between organisational contexts with greater independence, covering a broad range of concerns in the context of organisational autonomy. Goal-driven agents may act in a system containing several organisations, and *interact* with these organisations considering agent-internal goals. We claim that an agent that is aware of organisations (in general) and can hence better understand specific organisations it encounters at run time is able to reach higher autonomy. Such an agent would, for instance, know that another agent is deontically bound to performing a specific service, and that it may hence interact with that agent towards fulfilling a goal that requires this service. This is again similar to human interactions: Humans learn that they may trust a bus driver to share accurate information about bus routes. Crucially, they further know that this trust is modulated not by the bus driver (as an autonomous agent) but it is due to the bus driver being bound by the organisational principles of the public transport company they work for. In addition to interacting with organisations, agents might *join* organisations to then achieve an alignment of potentially conflicting knowledge or goals. That is, an agent's *decision to join* a specific organisation and to adopt a particular organisational role can be informed by knowledge that the agent obtains about the organisation, prior to joining.

13:10 Autonomy in the Age of Knowledge Graphs

Technically, this may be enabled through agents exchanging knowledge about organisations through KGs directly, or by making this information discoverable through environmental artifacts; the latter case also can be extended to permit (sufficiently authorised) agents to modify organisational knowledge at run time and hence allows groups of agents to evolve organisations together. Furthermore, KGs can provide agents that join an organisation with fundamental knowledge that is required for successful (inter-)action in the context of this organisation; i.e., knowledge about the available roles and objectives, and even knowledge that helps an agent decide about whether it should commit to a specific role in the organisation. To this end, KGs can provide deontic specifications of the organisation’s objectives, as well as of the policies and norms that agents must and should, respectively, comply with in order to reach the organisational objectives or agent-specific goals. Since these goals may conflict with organisational objectives and with the goals of other agents, knowledge about sanctioning mechanisms may also be shared, which again leads to an increase in an agent’s autonomy with respect to the organisation. Equipped with this knowledge, it can now make an informed decision about whether or not to break norms that are set by the organisation. Here, one can view the interaction of agents with the organisation (or with each other through the organisation) from a *governance* perspective: the organisation governs agent behaviour through norms and policies, and the organisation’s agents govern the organisation, e.g., by changing organisational policies or by gradually evolving norms. Then, KGs can provide a transparent representation not only of the objectives, policies, and norms, but also about the meta-framework that governs them, thus enabling agents to affect organisational change in a structured manner. With respect to technologies, constraints on KGs – which can be represented using languages such as SHACL – can play an important role for organisations of autonomous agents. For example: constraints that are to be satisfied in the future may model organisational goals; constraints that have to be satisfied amount to policies; and constraints that should typically be satisfied may represent norms.

3.4 Knowledge supports Internal Autonomy

So far, we have discussed how knowledge can support autonomy from an *external* perspective: an agent is typically situated in an *environment* (see Section 3.1) and interacts with artifacts and *other agents* (see Section 3.2), and potentially as part of an *organisation* (see Section 3.3). Across these dimensions, we argued that a formal, explicit, knowledge-level representation of relevant entities allows the agent to reason and decide about its interactions with the rest of the system.

We now finally survey the agent’s internal perspective, where autonomy relates to the agent’s freedom of *selecting the goals* it is working towards, and of *executing a course of action* for achieving selected goals. Following Castelfranchi and Falcone [17], an agent has *goal autonomy* if: i) it is endowed with its own goals (e.g., at design time); or ii) it adopts goals received from other agents only when the adopted goals enable some of its own goals. The latter relates to the discussion in Section 3.2 about *autonomy in delegation*, but from an internal perspective that focuses on the receiving agent’s ability to select its own goals. Goal autonomy hence is compatible with social ability: If an agent interacts with other agents and can adopt goals from them, then the agent is susceptible to influence – but the agent still preserves its goal autonomy if it can reason about the relations between candidate goals and its own goals, and if it can select specific goals to be adopted based on this reasoning. We argue that, similar to how KGs enable an agent to achieve autonomy with respect to the environment, other agents, and organisations, KGs may also help an agent in this very reasoning process, and hence help it *achieve* goal autonomy. KGs would do this by representing explicitly or implicitly relations between candidate goals and the agent’s own goals; they may also represent commonsense knowledge (e.g., see [42]), thereby providing a foundation for the agent’s reasoning about goal adoption.

After selecting a goal, an agent will typically strive to execute a course of action towards achieving that goal. To do this, it requires *procedural knowledge*, which can take various forms: For example, procedural knowledge can be defined in terms of *plans* that specify sequences of actions and are programmed by a developer. Conversely, within a marketplace, an auctioneer may need to declare formally the dimensions along which the *auction protocols* may vary (ascending or descending, sealed-bid or open-cry, single-shot or continuous, first or second price, etc.) to allow agents to reason whether or not they should engage in the auction, as well as providing evidence to bidders of the auction outcome [6]. In the case of a reinforcement learning agent, procedural knowledge takes the form of a *policy* that is learned through repeated interactions with the environment. In other systems, from a practical/industry-oriented perspective, procedural knowledge could be represented through *workflows, processes, or standard procedures* [5, 74]. Agents' goal achievement autonomy – that is, their executive autonomy – is increased if agents are able to acquire procedural knowledge *at run time*. However, the means to represent procedural knowledge depend on the internals of the agents using the knowledge. In some cases, KGs might provide a suitable solution for representing such knowledge themselves [88]. In the more general case, KGs may provide a uniform way of describing and linking to procedural knowledge such that agents can then acquire the knowledge in the context of their operation. We hence propose that KGs may not only directly enable executive autonomy, but that they may – across many scenarios – provide a uniform interface to procedural knowledge that itself is represented in diverse formats.

3.5 Knowledge Construction and Evolution

We propose that KGs provide a solid foundation for supporting autonomy at different levels, including environmental, social, and deontic aspects as well as an agent's internal goal-selection and execution. However, this knowledge is neither static nor immutable [24]: In the case of knowledge regarding the agents' environment, this may change over the course of time, and represent variations in the conditions and stimuli accessible to the agents. Individual and communal knowledge evolves according to the changes in the internal state of the agents, as well as according to the different interactions that they may engage in. Moreover, this knowledge is not simply available for querying as a set of *read-only* resources, but it is actually subject to modifications by the very autonomous agents that use it. Using KGs to represent this knowledge caters to these requirements: KGs – and the contained entities – feature unique identifiers and can hence be easily and unambiguously addressed and, if need be, versioned, in centralised as well as decentralised settings.

The way these KGs are constructed may differ in terms of approach and/or underlying technologies [89]: Individual autonomous agents might develop their own knowledge independently of their peers, where they will in the case of KGs typically use at least a common standard ontology or controlled vocabulary, in order to allow interoperability at the time of inter-agent interactions. On the other hand, if agents are limited to updating existing knowledge with new information (e.g., in the case of new observations from a sensor), the knowledge structure does not change often, but the observations may rapidly become stale [59]. This is often the case in IoT environments, where sensors and actuators rely on knowledge that takes the form of dynamic streams of data [79]. Agents need to react in a timely fashion to these data streams, so as to update their beliefs or revise their intentions. Although certain extensions exist for allowing KGs to integrate stream reasoning [26] and continuous querying capabilities [25, 51, 9], the usage of streaming KGs by autonomous entities is still incipient. Beyond traditional pipelines of data streams, the employment of streaming KGs in multi-agent systems and in general by decentralised autonomous entities has the potential to boost the exploitation of local and personal information captured by devices on the edge and other sensing agents.

13:12 Autonomy in the Age of Knowledge Graphs

Moreover, autonomous systems need to be able to represent and reason with knowledge about external entities (the environment, other agents, organisations) but also with internal knowledge about their goals, tasks and capabilities. The former is primarily knowledge that naturally emerges from heterogeneous agents cooperating to reach some common goal. Knowledge is hence created and exchanged following agent communication and social mechanisms [16]. This knowledge may emerge from the agreements and deliberations that arise during the lifetime of autonomous entities [55]. Ontologies may then be constructed from commonly learned facts contributed by autonomous agents, thus constituting an interaction-driven KG. Given that this knowledge is built from diverse and potentially contradicting sources, different methods of reconciliation and mediation have been explored; for example by relying on ontology alignment techniques [14, 31], or distributed ontology evolution [56].

In this kind of scenario, knowledge is heterogeneous and distributed, and therefore interaction mechanisms and protocols cannot assume that agents share common knowledge beyond what is required to establish their interactions. This heterogeneity also implies that agents can use both (onto)logical or, more broadly, symbolic knowledge as well as numerical, sub-symbolic knowledge representation. The evolution of this type of knowledge cannot happen in a siloed manner, since each type of knowledge representation is essential to cover different aspects; i.e., abstracting from and making sense of stimuli from the environment. Furthermore, autonomous and cooperative decision-making implies that all the systems inhabiting the environment have to respond to the same type of demands, therefore it does not make sense for them to react to these demands in isolation, especially when they are cooperating towards a common goal.

Any knowledge evolution model should hence consider the abilities of the autonomous agents to: i) reason and evolve their own knowledge, which may be modelled using disparate formalisms; ii) reason about the different demands they perceive from the environment they are immersed in; iii) deliberate about the knowledge to evolve, given conflicting demands (and ultimately knowledge); and iv) interact with other autonomous systems to communicate and cooperatively decide the best course of action.

4 KGs and Autonomy

Section 3 identifies how knowledge supports different dimensions of autonomy, with respect to the environment, other agents, organisations, and internally, regarding executive and goal-setting autonomy. This knowledge, as discussed in Section 3.5, is dynamically constructed and evolves as a result of decision making by autonomous systems that have a common goal. The use of KGs in autonomous systems is quite disparate, and KGs are often used to support only some aspects of autonomy, e.g., knowledge about the environment in W3C WoT TD, or about the capabilities of other autonomous systems, e.g., as in the MARIO reference architecture for social robots [4].

Following from the discussion in the previous section, we argue that all of these dimensions of autonomy can benefit from the use of KGs, where the use of Semantic Web languages such as RDFS and OWL overcomes syntactic interoperability issues and makes the vocabulary to use in communicative acts (and any constraints on its use) explicit.

In the remainder of this section we identify the characteristics that KGs should exhibit in order to enable the holistic view of autonomy we discussed so far: i.e., the expressivity of the representation language used to model the KG (in terms of the primitives and the consequent reasoning services) and the coverage of vocabulary used. These should promote autonomous behaviour by supporting the mechanisms used for interaction, governance with respect to organisational autonomy, and an explicit representation of how to perform a given task.

4.1 Expressivity and Coverage of KGs Supporting Autonomy

A recent systematic review [2] surveyed a number of knowledge-enabled approaches to develop and deploy autonomous robots in order to assess whether current state of the art ontology-based approaches model those concepts and relations that are necessary to represent knowledge about other autonomous entities and the environment; i.e., the ability to model and reason about the system’s own capabilities and the capabilities of others. This study argues that the key factors for the success of knowledge-based autonomous systems is the use of explicit ontological models to support autonomous systems in understanding and reasoning about their environment, their own internal knowledge and knowledge about other agents and organisations. The systematic review is based on Langley’s characterisation of autonomy with respect to abstract reasoning tasks such as recognition and categorisation, decision-making and choice, problem solving and planning, as well as perception and situation assessment [50]. Langley distinguishes between: i) *what*, i.e., the functional capabilities that an autonomous system should exhibit; and ii) the way in which these functional modules should interconnect (*how*). Functional capabilities and their interconnections determine the specific vocabulary needed to represent autonomy and the types of reasoning tasks that are supported by different ontological models. These interconnections are supported by the exchange of data and knowledge that is often both produced and consumed by autonomous systems.

KG schemata modelling autonomous behaviour should include the definition of entities (in terms of modules or fragments of hierarchical knowledge) that represent both static (e.g., *agent*, *object*, or *scene*) and dynamic (e.g., *task*, *goal*, or *action*) aspects of autonomy. We follow the same characterisation and we discuss the ways KGs are currently used to support autonomy, as a way to identify the limitations (which are introduced in Section 5).

As discussed in Section 2, there is no accepted definition of KG, and as a result the term KG is used to refer to different data sources, and for example, to model both real-time data generated by sensors that does not conform to some schema, or a graph of interrelated data that is modelled according to some explicit schemata that often includes standard foundational (upper-level) and domain-specific ontologies, as well as organisation-specific schemata (which might be proprietary). Foundational ontologies such as SUMO [62] or Dolce [35] and its simplified version in OWL, Dolce+DnS Ultralite⁶ are used to model domain-independent concepts, e.g., time, object or event, and support interoperability at the highest level of abstraction. Domain-specific vocabularies have emerged in many domains to provide common ground for communication and have become standards to model data for the specific domain. Examples of such standards are the Semantic Sensor Network (SSN) ontology⁷ [37], and the provenance ontology Prov-O⁸ [53]. SSN is a modular ontology that also includes a self-contained core ontology modelling Sensors, Observations, Samples, and Actuators (SOSA). SSN and SOSA can support a variety of applications and use cases, including satellite imagery, large-scale scientific monitoring, and use cases across the Web of Things [43]. PROV-O is an OWL ontology that describes the classes, properties and restrictions describing the provenance data model, used to integrate provenance information in different systems and based on different contexts. More recently, the W3C WoT TD [43] standard provides the vocabulary for describing the metadata and interfaces for a “Thing”, i.e., an abstraction of an entity (physical or virtual) that interacts with and participates in the Web of Things, thereby endowing agents with higher autonomy with respect to their environment (see Section 3.1).

⁶ <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl>

⁷ <https://www.w3.org/TR/2017/REC-vocab-ssn-20171019/>

⁸ <https://www.w3.org/TR/prov-o/>

4.2 KGs Supporting Autonomous Agents' Governance

Regarding organisational autonomy (see Section 3.3), a similar issue manifests itself when representing and reasoning with obligations and norm governance in general [44]. The ODRL standard⁹ is a formal language, expressed in RDF, for modelling policies regulating the use of digital rights. However, the ODRL core that models the components of a policy expression does not provide any formal specification of its semantics, and it provides only informal semantics described in English [81]. Recently, some extensions have been proposed, based on the work in Normative MAS, where autonomy determines the decision-making processes [20]. For example, Fornara and Colombetti [33] aim to fully capture the formalisation of norms used to express obligations, permissions and prohibitions by modelling the operational semantics of conditional obligations using OWL, possibly extended with SWRL [39] rules to overcome OWL's limited expressivity with respect to the state of an interaction and the rules for computing the state of obligations [33]. SWRL adds Horn-style rules to OWL, thus permitting the modelling of additional domain knowledge that cannot be done by OWL alone. Both OWL and SWRL adopt the Open World assumption, whereby the absence of a statement does not necessarily mean that it can be inferred to be true or false. This is in contrast with SHACL, which adopts the Closed World Assumption (a statement that is not known to be true must be false). More recently, SHACL has been proposed to model more complex type of norms, such as compensatory norms [75].

4.3 KGs Supporting Agent Interaction

Several conceptual architectures and programming platforms that support autonomous behaviour are currently available (e.g., JaCaMo [12] or KnowRob [85]), and they rely on domain and foundational ontologies and KGs for representing and sharing declarative knowledge. However, many of the reasoning tasks underlying executive autonomy (see Section 3.4) involve procedural knowledge that is not typically captured by KGs. The execution of complex tasks involving several autonomous entities in open environments requires them to interact with each other and the environment, and to coordinate their executions; e.g., to support autonomy in delegation. Interaction protocols (also called *choreographies* within the service community) are formal specifications of these interactions, which describe the roles played by the autonomous systems, the allowed actions and choices, and the states and termination conditions [8]. Similarly, communication protocols specify the rules of interactions governing a dialogue between autonomous systems [29, 34, 71]. Typically, these protocols are expressed in some protocol language based on different abstractions and operational assumptions that are difficult to compare. An exhaustive description of these protocols is outside the scope of this paper, but we refer to the work by Chopra and colleagues for a review of the main formalisms and a comparison framework [21]. These protocols ground the meaning of the different concepts involved in some ontology, therefore implicitly establishing an a-priori commitment to a shared understanding of the terms being communicated [41], where the procedural part of the model and the execution traces are typically not included in KGs.

4.4 How to: KGs for Supporting Agents in Performing Tasks

Various efforts in robotics have also attempted to model and execute procedural knowledge about some tasks, e.g., *kitting*, where a robot places (a set of) parts on a tray and carries them to an assembly cell, using a knowledge-enabled approach. For example, the KnowRob [85] architecture has been extended with OWL based assembly planning, where assembled products are modelled

⁹ <https://www.w3.org/TR/odrl-model/>

according to ontologies, which are then used to plan the next action according to faulty and missing assertions in the robot’s beliefs about an ongoing assembly task [11]. Other knowledge-enabled approaches, however, convert OWL descriptions into PDDL [58] specifications [7, 47].

Multi-dimensional event knowledge graphs (EKGs) [30] have been proposed as a way to represent process events in a way that supports the execution and inspection of process traces. Recently, EKGs have also been proposed to model process logs [45] in order to support process analysis.

KGs have also been explored as a possible source of knowledge to be injected in reinforcement learning approaches [1, 3], where autonomous agents use sequential decisions to learn and adapt through interactions with their environment by trying to find (near-) optimal policies to perform an intended task [84]. In these approaches, commonsense KGs such as Wikidata [90] or CSKG, the Commonsense Knowledge Graph [42] are exploited to model commonsense notions such as spatial, part-whole, or temporal relations, and even affordances [19]. In line with our discussion in Section 3.5, we propose that such usage of KGs represents only the beginning of a movement to the more active curation of KGs *by* autonomous software agents themselves.

Recently, both KGs and architectures for autonomous agents have emerged as promising facilitators of autonomy in the context of generative AI and specifically Large Language Models (LLMs) [65, 82]. In this context, KGs are typically envisioned as providers of reliable, “hard” knowledge that LLMs are not capable of producing. While this emphasises the relevance and timeliness of research on KGs as facilitators of autonomy, we deliberately choose to present perspectives in this paper that can be considered orthogonal to LLMs and generative AI in order to avoid a tunnel view that is focused on current hype topics.

5 Limitations of Knowledge Graphs for Supporting Autonomy

Whilst there are successful examples of KGs that facilitate autonomy across the different dimensions of autonomy, employing KGs as facilitators of autonomy may bring substantial challenges with it, depending on context and scale. This section describes where KGs are in principle not suitable or are not sufficient to support autonomy, as well as where they are not *yet* suitable. From the resulting list of limitations, we identify challenges that exist today, and provide initial sketches of how these challenges could be addressed. We claim that the following factors can limit the applicability of KGs as facilitators of autonomy:

1. **Dynamics of participating systems.** Autonomous software systems can be expected to appear in highly dynamic environments, and they may frequently change, often without direct human involvement (cf. Section 4.4). Consequently, KGs need to be able to accommodate and adapt to frequent changes in the knowledge they provide; otherwise, knowledge consumers risk basing their decisions and actions on stale or deprecated knowledge that no longer reflects the environment they act in. A key question in this context is to what extent KGs can maintain models of dynamic behaviour as first-class abstractions, making it easier to adjust to change and to mitigate the resulting uncertainty.
2. **Scale of knowledge.** The content of knowledge-based systems (such as KGs) is notoriously effortful to maintain. While the maintenance issues on schema-level are alleviated by (schema-free) KGs (in contrast to traditional knowledge-based systems), the challenge of maintaining the contained knowledge at scale remains. Specifically, a common challenge – in addition to stale knowledge, as mentioned above – is that ensuring consistency will become an increasingly tedious task with growing KGs. Currently, it is not clear which use cases are favourable to large-scale knowledge maintenance of KG content and which are not. Our assumption is that given current practices, not all potential use cases of KGs for autonomous agents are viable

13:16 Autonomy in the Age of Knowledge Graphs

at scale and that viability can depend on a number of factors, such as the economic and technical resources available for knowledge maintenance and the extent to which the knowledge is open and considered generally useful beyond a specific target use case. For example, we expect that open knowledge that is managed by long-lived communities is more scalable than domain-specific knowledge that remains in silos. On a per-case basis, the maintainability of a KG will partially depend on the selected technologies and hence on the diversity and maturity of the KG technology ecosystem (cf. Section 4.1).

3. **Knowledge maintenance by humans and machines.** A key challenge that might, if overcome, mitigate the scalability issue mentioned above, is to have knowledge maintained jointly by humans and machines, serving the needs of both (as hinted at in Section 4.4). In particular, knowledge that machines can efficiently consume may be hard to interpret for humans, and vice versa. At the same time, duplication of knowledge should be avoided and joint knowledge maintenance can facilitate efficient knowledge maintenance, if accessibility and interpretability of both humans and machines can be guaranteed.
4. **Heterogeneity of participating systems.** A Web of autonomous agents will in practice be heterogeneous from architectural and technological perspectives; a substantial part of this heterogeneity will stem from legacy software that does not follow the idiomatic principles of conceptual meta-models (cf. Section 4.3) that will ideally emerge for future Web-based systems. These legacy software systems will not be able to utilise KGs in a comprehensive manner and their ability to be active and passive facilitators of autonomy will be limited: without tight coupling, these legacy systems will not be able to explore the Web, nor will they be straightforwardly discoverable by others. Also, interoperability issues can be expected to render governance (cf. Section 4.2) of legacy software-heavy agents a difficult task. Proposals for facilitating autonomy must account for such limitations, i.e., they must not assume that all systems relevant in a given application context are “good citizens” given idealistic architectural requirements.
5. **Knowledge modelling and representation choices.** KGs allow for a plethora of ways to model knowledge (cf. Section 4.1). However, inappropriate knowledge modelling may over-complicate KG usage. For example, certain ontologies are too detailed and too complicated for use cases that do not benefit from the level of precision and granularity that these ontologies provide. An agent specialised for working in a domain such as healthcare may benefit from access to the corresponding ontologies; for simpler use cases, it may be better to merely expose a subset of the knowledge in a more straightforward manner, and hence to modularise the underlying knowledge [64]. Towards addressing the same challenge of real-world applicability, the KG community should consider more actively managing interfaces to systems that (for organisational or technological reasons) are not willing to adopt elements of the Semantic Web stack, and instead prefer shallow representations such as plain JSON. Such systems can still actively participate in a global knowledge organisation as envisioned in this paper, but the interfaces need to be maintained.
6. **Modelling of procedural knowledge.** As mentioned in Section 3.4, the provision of procedural knowledge is key to enabling and augmenting autonomous behaviour. However, while several works propose and study the management of procedural knowledge with KGs (cf. Subsection 4.4), KGs do not provide first-class abstractions of procedural knowledge; and in industry applications, the management of procedural knowledge such as business process specification on the one hand and the maintenance of domain models in KGs and ontologies on the other are seen as orthogonal tasks. Integration of these types of knowledge is still an understudied challenge.

Given the challenges presented above, we identify the following research directions that can help mitigate them, although each of them will require substantial further investigation.

Hybrid representation of knowledge. The flexible nature of KGs allows for the integration of data in various ways that does not reflect the graph structure; e.g., knowledge does not necessarily need to be represented in a particular, rigid manner (think of RDF triples), but may as well be embedded as literals or be referenced through *Internationalized Resource Identifiers (IRIs)*, as shown for example in [60]. Such flexibility mitigates the unsuitability of KGs to certain kind of information, such as procedural knowledge and constraints (Challenge 6) and allows for nuanced and pragmatic modelling decisions (Challenge 5), thus facilitating KG maintenance at scale (Challenge 2). Still, open questions remain regarding the trade-offs that need to be made when compromising on knowledge representation (from a KG perspective), and when implementing reasoning and decision-making capabilities. With these trade-offs, one can properly interpret external knowledge or knowledge represented as literals, which essentially requires lifting basic knowledge representation and reasoning tasks to the business logic layer.

Ontology summarisation. Knowledge in KGs may not be particularly accessible for some agents, either because it is hard to find given the agents' limitations (Challenge 4), or because it is not modelled on the abstraction level that the agent requires (Challenge 5). Ontology summarisation approaches [86] can be applied to extract concise aggregations of knowledge that is relevant in a given context from a KG, thus making the content in a KG easier to consume for humans or machines.

Modularity of knowledge. In many scenarios, agents may only be able to work with a sub-graph of a given KG, either because of partial semantic interoperability or partial agreement and trust [27]. It can therefore make sense to take a modular view on KGs, for example by modularising knowledge into run-time-loadable “knowledge packs” such as proposed in [57]. In such systems, an individual agent may decide to work only with the sub-graph it can understand and considers reliable in the broader sense, whereas a group of agents may agree on a sub-graph as a joint ground truth. Thus, modular KGs can facilitate applicability in the context of heterogeneous, partially interoperable sub-systems (Challenges 4 and 5) and account for heterogeneity of content, e.g., regarding the frequency of updates (Challenge 1) and the general state of maintenance (Challenge 3).

(Semi-)Automating knowledge graph maintenance. To increase the feasibility of large-scale KGs (Challenge 2), their maintenance can be partially automated by utilising software agents of various degrees of autonomy and technical sophistication. Agents in the broader sense – i.e., simple bots for information crawling and the propagation of simple updates – already play a role in maintaining KGs today. Thus, social machines of humans and artificial agents that jointly maintain knowledge could profit from the use of organisational abstractions from multi-agent systems research, as shown in [91]. How to best advance the joint human-machine maintenance of knowledge is, however, an open question (Challenge 3), which must be carefully approached considering current hypes and trends: the emergence of LLM-based bots and the resulting threat of low-quality, technically wrong, intransparent, or otherwise undesirable “knowledge” [10] makes structured approaches to knowledge management and maintenance even more important.

6 Conclusions

In open systems, the management of knowledge is a crucial enabler of autonomous interaction among autonomous agents, including humans and software systems. We provide a conceptual vision of how KGs can be key facilitators of such autonomy. We propose that KGs enable the programming of artificial agents at the knowledge level – thus alleviating the need for low-level hard-wiring across the dimensions of autonomy – and discuss their merit with a special focus on the Web as an environment where interactions between autonomous agents take place. Based on an overview of current and fundamental limitations of KGs, we furthermore outline a set of challenges affecting the use of KGs as facilitators of autonomy; in particular, it is crucial to enable the joint human-machine maintenance of knowledge in highly dynamic and heterogeneous Web-scale systems. We then briefly discuss potential approaches, such as hybrid and modular knowledge representation, to address the challenges. Overall, this paper informs cross-community efforts to move towards reliable and resilient autonomous systems at Web scale.

References

- 1 Andrea Agiollo, Andrea Rafanelli, Matteo Magnini, Giovanni Ciatto, and Andrea Omicini. Symbolic knowledge injection meets intelligent agents: QoS metrics and experiments. *Autonomous Agents and Multi-Agent Systems*, 37(2), jun 2023. doi: 10.1007/S10458-023-09609-6.
- 2 Alberto Olivares Alarcos, Daniel Beßler, Alaa M. Khamis, P-J. Sequeira Gonçalves, Maki K. Habib, Julita Bermejo-Alonso, Marcos E. Barreto, Mohammed Diab, Jan Rosell, João Quintas, Joanna Isabelle Olszewska, Hirenkumar Chandrakant Nakawala, Edison Pignaton de Freitas, Amelie Gyrard, Stefano Borgo, G. Alenyà, Michael Beetz, and Howard Li. A review and comparison of ontology-based approaches to robot autonomy. *The Knowledge Engineering Review*, 34, 2019. doi:10.1017/S0269888919000237.
- 3 Prithviraj Ammanabrolu and Matthew J. Hausknecht. Graph constrained reinforcement learning for natural language action spaces. 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=B1x6w0EtwH>.
- 4 Luigi Asprino, Paolo Ciancarini, Andrea Giovanni Nuzzolese, Valentina Presutti, and Alessandro Russo. A reference architecture for social robots. *Journal of Web Semantics*, 72:100683, 2022. doi:10.1016/J.WEBSEM.2021.100683.
- 5 Stefan Bachhofner, Elmar Kiesling, Kate Revoredo, Philipp Waibel, and Axel Polleres. Automated process knowledge graph construction from BPMN models. In Christine Strauss, Alfredo Cuzzocrea, Gabriele Kotsis, A. Min Tjoa, and Ismail Khalil, editors, *Database and Expert Systems Applications*, pages 32–47, Cham, 2022. Springer International Publishing. doi:10.1007/978-3-031-12423-5_3.
- 6 Wei Bai, Emmanuel Tadjouddine, and Terry Payne. A dialectical approach to enable decision making in online trading. In Michael Rovatsos, George Vouros, and Vicente Julian, editors, *Multi-Agent Systems and Agreement Technologies*, pages 203–218. Springer International Publishing, 2016. doi: 10.1007/978-3-319-33509-4_17.
- 7 Stephen Balakirsky, Zeid Kootbally, Thomas Kramer, Anthony Pietromartire, Craig Schlenoff, and Satyandra Gupta. Knowledge driven robotics for kitting applications. *Robotics and Autonomous Systems*, 61(11):1205–1214, 2013. Ubiquitous Robotics. doi:10.1016/J.ROBOT.2013.04.006.
- 8 Matteo Baldoni, Cristina Baroglio, Amit K. Chopra, Nirmitt Desai, Viviana Patti, and Munindar P. Singh. Choice, interoperability, and conformance in interaction protocols and service choreographies. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2009*, volume 2, pages 843–850. International Foundation for Autonomous Agents and Multiagent Systems, 2009. URL: <https://dl.acm.org/doi/abs/10.5555/1558109.1558129>.
- 9 Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. C-SPARQL: a continuous query language for RDF data streams. *International Journal of Semantic Computing*, 4(01):3–25, 2010. doi: 10.1142/S1793351X10000936.
- 10 Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21*, pages 610–623, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3442188.3445922.
- 11 Daniel Beßler, Mihai Pomarlan, and Michael Beetz. OWL-Enabled Assembly Planning for Robotic Agents. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 1684–1692. International Foundation for Autonomous Agents and Multiagent Systems, 2018. URL: <https://dl.acm.org/doi/10.5555/3237383.3237950>.
- 12 Olivier Boissier, Rafael H. Bordini, Jomi Fred Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with JaCaMo. *Sci-*

- ence of Computer Programming, 78(6):747–761, 2013. doi:10.1016/J.SCICO.2011.10.004.
- 13 Rafael H. Bordini, Amal El Fallah Seghrouchni, Koen Hindriks, Brian Logan, and Alessandro Ricci. Agent programming in the cognitive era. *Autonomous Agents and Multi-Agent Systems*, 34(2), may 2020. doi:10.1007/S10458-020-09453-Y.
 - 14 Yasser Bourahla, Manuel Atencia, and Jérôme Euzenat. Knowledge improvement and diversity under interaction-driven adaptation of learned ontologies. In *AAMAS '21: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 242–250. International Foundation for Autonomous Agents and Multiagent Systems, 2021. URL: <https://dl.acm.org/doi/10.5555/3463952.3463986>, doi: 10.5555/3463952.3463986.
 - 15 Dan Brickley and Ramanathan Guha. RDF Schema 1.1. W3C recommendation, World Wide Web Consortium, feb 2014. URL: <https://www.w3.org/TR/2014/REC-rdf-schema-20140225/>.
 - 16 John Cantwell. A formal model of multi-agent belief-interaction. *Journal of Logic, Language and Information*, 15:303–329, 2006. doi:10.1007/S10849-006-3776-3.
 - 17 Cristiano Castelfranchi and Rino Falcone. *From Automaticity to Autonomy: The Frontier of Artificial Agents*, pages 103–136. Springer US, Boston, MA, 2003. doi:10.1007/978-1-4419-9198-0_6.
 - 18 Adam Cheyer and David Martin. The Open Agent Architecture. *Autonomous Agents and Multi-Agent Systems*, 4(1–2):143–148, 2001. doi: 10.1023/A:1010091302035.
 - 19 Prateek Chhikara, Jiarui Zhang, Filip Ilievski, Jonathan Francis, and Kaixin Ma. Knowledge-enhanced Agents for Interactive Text Games, 2023. doi:10.48550/arXiv.2305.05091.
 - 20 Amit Chopra, Leendert van der Torre, Harko Verhagen, and Serena Villata. *Handbook of Normative Multiagent Systems*. College Publications, Rickmansworth, 2018.
 - 21 Amit K. Chopra, Samuel H. Christie V, and Munindar P. Singh. An evaluation of communication protocol languages for engineering multiagent systems. *Journal of Artificial Intelligence Research*, 69:1351–1393, dec 2020. doi:10.1613/JAIR.1.12212.
 - 22 Richard Cyganiak, David Wood, and Markus Lanthaler. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation, World Wide Web Consortium, February 25 2014. URL: <https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>.
 - 23 Marina De Vos, Sabrina Kirrane, Julian Padget, and Ken Satoh. ODRL Policy Modelling and Compliance Checking. In Paul Fodor, Marco Montali, Diego Calvanese, and Dumitru Roman, editors, *Rules and Reasoning*, pages 36–51, Cham, 2019. Springer International Publishing. doi: 10.1007/978-3-030-31095-0_3.
 - 24 Emanuele Della Valle, Stefano Ceri, Frank Van Harmelen, and Dieter Fensel. It’s a streaming world! reasoning upon rapidly changing information. *IEEE Intelligent Systems*, 24(6):83–89, 2009. doi:10.1109/MIS.2009.125.
 - 25 Daniele Dell’Aglia, Jean-Paul Calbimonte, Emanuele Della Valle, and Oscar Corcho. Towards a Unified Language for RDF Stream Query Processing. In *The Semantic Web: ESWC 2015 Satellite Events - ESWC 2015 Satellite Events Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*, pages 353–363. Springer, 2015. doi:10.1007/978-3-319-25639-9_48.
 - 26 Daniele Dell’Aglia, Emanuele Della Valle, Frank van Harmelen, and Abraham Bernstein. Stream reasoning: A survey and outlook. *Data Science*, 1(1–2):59–83, 2017. doi:10.3233/DS-170006.
 - 27 Paul Doran, Valentina Tamma, Terry R. Payne, and Ignazio Palmisano. Dynamic selection of ontological alignments: a space reduction mechanism. In *Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09) (10/07/09 - 16/07/09)*, jul 2009. URL: <http://ijcai.org/Proceedings/09/Papers/334.pdf>.
 - 28 Lisa Ehrlinger and Wolfram Wöß. Towards a definition of knowledge graphs. In *International Conference on Semantic Systems*, 2016. URL: <https://ceur-ws.org/Vol-1695/paper4.pdf>.
 - 29 Ulrich Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni. Aspects of protocol conformance in inter-agent dialogue. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 982–983, 2003. doi:10.1145/860575.860756.
 - 30 Stefan Esser and Dirk Fahland. Multi-dimensional event data in graph databases. *Journal on Data Semantics*, 10:109–141, 2020. doi:10.1007/S13740-021-00122-1.
 - 31 Jérôme Euzenat. Interaction-based ontology alignment repair with expansion and relaxation. In *IJCAI 2017-26th International Joint Conference on Artificial Intelligence*, pages 185–191. AAAI Press, 2017. doi:10.24963/IJCAI.2017/27.
 - 32 Jacques Ferber and Olivier Gutknecht. A meta-model for the analysis and design of organizations in multi-agent systems. In Yves Demazeau, editor, *Proceedings of the Third International Conference on Multiagent Systems, ICMAS 1998, Paris, France, July 3-7, 1998*, pages 128–135. IEEE Computer Society, 1998. doi:10.1109/ICMAS.1998.699041.
 - 33 Nicoletta Fornara and Marco Colombetti. Using semantic web technologies and production rules for reasoning on obligations, permissions, and prohibitions. *Ai Communications*, 32(4):319–334, 2019. doi:10.3233/AIC-190617.
 - 34 Foundation for Intelligent Physical Agents. FIPA ACL Message Structure Specification. Technical report, Foundation for Intelligent Physical Agents – FIPA ACL Message Structure Specification, 2002. URL: <http://www.fipa.org/specs/fipa00061/SC00061G.pdf>.
 - 35 Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with DOLCE. In *International conference on knowledge engineering and knowledge management*, pages 166–181. Springer, 2002. doi:10.1007/3-540-45810-7_18.
 - 36 Amélie Gyrard, Pankesh Patel, Amit Sheth, and Martin Serrano. Building the web of knowledge with smart IoT applications. *IEEE Intelligent Systems*, 31(5):83–88, 2016. doi:10.1109/MIS.2016.81.

- 37 Armin Haller, Krzysztof Janowicz, Simon J. D. Cox, Maxime Lefrançois, Kerry L. Taylor, Danh Le-Phuoc, Joshua Lieberman, Raúl García-Castro, Robert Atkinson, and Claus Stadler. The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web*, 10:9–32, 2018. doi:10.3233/SW-180320.
- 38 Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d’Amato, Gerard de Melo, Claudio Gutiérrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. *Knowledge Graphs*. Number 22 in Synthesis Lectures on Data, Semantics, and Knowledge. Springer, 2021. doi:10.2200/S01125ED1V01Y202109DSK022.
- 39 Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, S. Tabet, Benjamin Grosz, and Mike Dean. SWRL: A semantic web rule language combining OWL and RuleML. Technical report, W3C Member Submission, 2004. URL: <https://www.w3.org/submissions/SWRL/>.
- 40 Thomas Hubauer, Steffen Lamparter, Peter Haase, and Daniel Markus Herzig. Use cases of the industrial knowledge graph at Siemens. In Marieke van Erp, Medha Atre, Vanessa López, Kavitha Srinivas, and Carolina Fortuna, editors, *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, October 8th - to - 12th, 2018*, volume 2180 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018. URL: <https://ceur-ws.org/Vol-2180/paper-86.pdf>.
- 41 Michael N. Huhns and Munindar P. Singh. Ontologies for agents. *IEEE Internet computing*, 1(6):81–83, 1997. doi:10.1109/4236.643942.
- 42 Filip Ilievski, Pedro Szekely, and Bin Zhang. CSKG: The commonsense knowledge graph. In *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18*, pages 680–696. Springer, 2021. doi:10.1007/978-3-030-77385-4_41.
- 43 Sebastian Käbisch, Ege Korkan, and Michael McCool. Web of things (WoT) thing description 1.1. W3C proposed recommendation, World Wide Web Consortium, jul 2023. URL: <https://www.w3.org/TR/2023/PR-wot-thing-description11-20230711/>.
- 44 Timotheus Kampik, Adnane Mansour, Olivier Boissier, Sabrina Kirrane, Julian A. Padget, Terry R. Payne, Munindar P. Singh, Valentina A. M. Tamma, and Antoine Zimmermann. Governance of autonomous agents on the web: Challenges and opportunities. *ACM Trans. Internet Techn.*, 22(4):104:1–104:31, 2022. doi:10.1145/3507910.
- 45 Shahrzad Khayatbashi, Olaf Hartig, and Amin Jalali. Transforming event knowledge graph to object-centric event logs: A comparative study for multi-dimensional process analysis. In *Proceedings of the 42nd International Conference on Conceptual Modeling*, pages 220–238. Springer Nature Switzerland, 2023. doi:10.1007/978-3-031-47262-6_12.
- 46 Holger Knublauch and Dimitris Kontokostas. Shapes Constraint Language (SHACL), W3C Recommendation 20 July 2017. W3c recommendation, World Wide Web Consortium, July 20 2017. URL: <https://www.w3.org/TR/2017/REC-shacl-20170720/>.
- 47 Zeid Kootbally, Craig I. Schlenoff, Christopher R. Lawler, Thomas R. Kramer, and Satyandra K. Gupta. Towards robust assembly with knowledge representation for the planning domain definition language (PDDL). *Robotics and Computer-Integrated Manufacturing*, 33:42–55, 2015. Special Issue on Knowledge Driven Robotics and Manufacturing. doi:10.1016/j.rcim.2014.08.006.
- 48 Markus Krötzsch, Frantisek Simancik, and Ian Horrocks. A description logic primer, 2013. doi:10.48550/arXiv.1201.4089.
- 49 Loredana Laera, Ian Blacoe, Valentina Tamma, Terry Payne, Jérôme Euzenat, and Trevor Bench-Capon. Argumentation over ontology correspondences in mas. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, 2007. doi:10.1145/1329125.1329400.
- 50 Pat Langley, John E. Laird, and Seth Rogers. Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, 10(2):141–160, 2009. doi:10.1016/J.COGLYS.2006.07.004.
- 51 Danh Le-Phuoc, Minh Dao-Tran, Josiane Xavier Parreira, and Manfred Hauswirth. A native and adaptive approach for unified processing of linked streams and linked data. In *International Semantic Web Conference*, pages 370–388. Springer, 2011. doi:10.1007/978-3-642-25073-6_24.
- 52 Danh Le-Phuoc, Hoan Nguyen Mau Quoc, Hung Ngo Quoc, Tuan Tran Nhat, and Manfred Hauswirth. The Graph of Things: A step towards the Live Knowledge Graph of connected things. *Journal of Web Semantics*, 37–38:25–35, 2016. doi:10.1016/J.WEBSEM.2016.02.003.
- 53 Tim Lebo, Satya Sahoo, Deborah McGuinness, Khalid Belhajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, and Jun Zhao. PROV-O: The PROV ontology: W3C recommendation 30 April 2013, apr 2013. URL: <https://www.w3.org/TR/prov-o/>.
- 54 Jérémy Lemée, Danai Vachtsevanou, Simon Mayer, and Andrei Ciortea. Signifiers for Affordance-driven Multi-Agent Systems. In *International Workshop on Engineering Multi-Agent Systems (EMAS) at the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2022. URL: <https://www.alexandria.unisg.ch/handle/20.500.14171/108730>.
- 55 Michael Luck and Peter McBurney. Computing as interaction: Agent and agreement technologies. In *IEEE International Conference on Distributed Human-Machine Systems*. IEEE Press, 2008. URL: <https://nms.kcl.ac.uk/michael.luck/resources/dhms08.pdf>.
- 56 Alexander Maedche, Boris Motik, Ljiljana Stojanovic, Rudi Studer, and Raphael Volz. An infrastructure for searching, reusing and evolving distributed

- ontologies. In *Proceedings of the 12th international conference on World Wide Web*, pages 439–448, 2003. doi:10.1145/775152.775215.
- 57 Simon Mayer, Jack Hodges, Dan Yu, Mareike Kritzler, and Florian Michahelles. An open semantic framework for the industrial internet of things. *IEEE Intelligent Systems*, 32(1):96–101, 2017. doi:10.1109/MIS.2017.9.
- 58 Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. PDDL – The Planning Domain Definition Language. Technical Report CVC TR98003/DCS TR1165, Yale Center for Computational Vision and Control, oct 1998. URL: <https://www.cs.cmu.edu/afs/cs/user/mmv/www/papers/98aips-PDDL.pdf>.
- 59 Sascha Meckler, Harald Steinmüller, and Andreas Harth. Building a knowledge graph with inference for a production machine using the web of things standard. In *Advances and Trends in Artificial Intelligence. From Theory to Practice: 34th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2021, Kuala Lumpur, Malaysia, July 26–29, 2021, Proceedings, Part II 34*, pages 240–251. Springer, 2021. doi:10.1007/978-3-030-79463-7_20.
- 60 Katarina Milenkovic, Simon Mayer, Konrad Diwold, and Josef Zehetner. Enabling knowledge management in complex industrial processes using semantic web technology, 2019. URL: <https://www.alexandria.unisg.ch/handle/20.500.14171/98451>.
- 61 Justin J. Miller. Graph Database Applications and Concepts with Neo4j. In *Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA March 23rd-24th, 2013*, pages 141–147. Association for Information Systems, 2013. URL: <https://aisel.aisnet.org/sais2013/24>.
- 62 Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9, 2001. doi:10.1145/505168.505170.
- 63 Pablo Noriega. *Agent Mediated Auctions: The Fishmarket Metaphor*. PhD thesis, Universitat Autònoma de Barcelona, 1997. URL: https://www.iiia.csic.es/research/thesis-details?pastphd_id=85.
- 64 Ignazio Palmisano, Valentina Tamma, Terry Payne, and Paul Doran. Task oriented evaluation of module extraction techniques. In Abraham Bernstein, David R. Karger, Tom Heath, Lee Feigenbaum, Diana Maynard, Enrico Motta, and Krishnaprasad Thirunarayan, editors, *The Semantic Web - ISWC 2009*, pages 130–145, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. doi:10.1007/978-3-642-04930-9_9.
- 65 Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeljanenko, Wen Zhang, Matteo Lissandrini, Russa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and Damien Graux. Large language models and knowledge graphs: Opportunities and challenges, 2023. doi:10.48550/arXiv.2308.06374.
- 66 Massimo Paolucci, Takahiro Kawamura, Terry R Payne, and Katia Sycara. Semantic matching of web services capabilities. In *The Semantic Web—ISWC 2002: First International Semantic Web Conference Sardinia, Italy, June 9–12, 2002 Proceedings 1*, pages 333–347. Springer, 2002. doi:10.1007/3-540-48005-6_26.
- 67 Massimo Paolucci, Onn Shehory, Katia Sycara, Dirk Kalp, and Anandee Pannu. A planning component for RETSINA agents. In Nicholas R. Jennings and Yves Lespérance, editors, *Intelligent Agents VI. Agent Theories, Architectures, and Languages*, pages 147–161, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. doi:10.1007/10719619_11.
- 68 Paolo Paretì and George Konstantinidis. A review of SHACL: From data validation to schema reasoning for RDF graphs. In *Reasoning Web. Declarative Artificial Intelligence: 17th International Summer School 2021, Leuven, Belgium, September 8–15, 2021, Tutorial Lectures*, pages 115–144, Berlin, Heidelberg, 2021. Springer-Verlag. doi:10.1007/978-3-030-95481-9_6.
- 69 Bijan Parsia, Peter Patel-Schneider, and Boris Motik. OWL 2 web ontology language structural specification and functional-style syntax (second edition). W3C recommendation, World Wide Web Consortium, dec 2012. <https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/>.
- 70 Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semant. Web*, 8(3):489–508, jan 2017. doi:10.3233/SW-160218.
- 71 Jeremy Pitt and Abe Mamdani. A protocol-based semantics for an agent communication language. In *IJCAI*, volume 99, pages 486–491, 1999. URL: <https://dl.acm.org/doi/abs/10.5555/1624218.1624288>.
- 72 David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526, 1978. doi:10.1017/S0140525X00076512.
- 73 Alessandro Ricci, Andrea Omicini, Mirko Viroli, Luca Gardelli, and Enrico Oliva. Cognitive stigmery: Towards a framework based on agents and artifacts. In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, *Environments for Multi-Agent Systems III*, volume 4389, pages 124–140, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. doi:10.1007/978-3-540-71103-2_7.
- 74 Dennis M. Riehle, Sven Jannaber, Patrick Delfmann, Oliver Thomas, and Jörg Becker. Automatically annotating business process models with ontology concepts at design-time. In Sergio de Cesare and Ulrich Frank, editors, *Advances in Conceptual Modeling*, pages 177–186, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-70625-2_17.
- 75 Livio Robaldo, Sotirios Batsakis, Roberta Calegari, Francesco Calimeri, Megumi Fujita, Guido Governatori, Maria Concetta Morelli, Francesco Paccenza, Giuseppe Pisano, Ken Satoh, Ilias Tachmazidis, and Jessica Zangar. Compliance check-

- ing on first-order knowledge with conflicting and compensatory norms: a comparison among currently available technologies. *Artificial Intelligence and Law*, jun 2023. doi:10.1007/s10506-023-09360-z.
- 76 Daniel Schraudner. Stigmergic multi-agent systems in the semantic web of things. In Ruben Verborgh, Anastasia Dimou, Aidan Hogan, Claudia d'Amato, Ilaria Tiddi, Arne Bröring, Simon Mayer, Femke Ongenaes, Riccardo Tommasini, and Mehwish Alam, editors, *The Semantic Web: ESWC 2021 Satellite Events*, pages 218–229, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-80418-3_34.
- 77 Guus Schreiber and Fabien Gandon. RDF 1.1 XML syntax. W3C recommendation, World Wide Web Consortium, feb 2014. URL: <https://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/>.
- 78 Paul R. Smart and Nigel R. Shadbolt. *Social Machines*, pages 6855–6862. IGI Global, jul 2015. doi:10.4018/978-1-4666-5888-2.ch675.
- 79 John Soldatos, Nikos Kefalakis, Manfred Hauswirth, Martin Serrano, Jean-Paul Calbimonte, Mehdi Riahi, Karl Aberer, Prem Prakash Jayaraman, Arkady Zaslavsky, Ivana Podnar Žarko, et al. Openiot: Open source internet-of-things in the cloud. In *Interoperability and Open-Source Solutions for the Internet of Things: International Workshop, FP7 OpenIoT Project, Held in Conjunction with SoftCOM 2014, Split, Croatia, September 18, 2014, Invited Papers*, pages 13–25. Springer, 2015. doi:10.1007/978-3-319-16546-2_3.
- 80 Sebastian Stein, Terry R. Payne, and Nicholas R. Jennings. Flexible provisioning of web service workflows. *ACM Trans. Internet Technol.*, 9(1):1–45, feb 2009. doi:10.1145/1462159.1462161.
- 81 Simon Steyskal and Axel Polleres. Towards formal semantics for ODRL policies. In Nick Bassiliades, Georg Gottlob, Fariba Sadri, Adrian Paschke, and Dumitru Roman, editors, *Rule Technologies: Foundations, Tools, and Applications*, pages 360–375, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-21542-6_23.
- 82 Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents, 2023. doi:10.48550/ARXIV.2309.02427.
- 83 Yousouf Taghzouti, Danai Vachtsevanou, Simon Mayer, and Andrei Ciortea. A step toward semantic content negotiation. In *23rd International Conference on Knowledge Engineering and Knowledge Management (EKAW)*. CEUR-WS, 2022. URL: <https://ceur-ws.org/Vol-3256/paper5.pdf>.
- 84 Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *J. Mach. Learn. Res.*, 10:1633–1685, dec 2009. doi:10.5555/1577069.1755839.
- 85 Moritz Tenorth and Michael Beetz. KnowRob: A knowledge processing infrastructure for cognition-enabled robots. *The International Journal of Robotics Research*, 32(5):566–590, 2013. doi:10.1177/0278364913481635.
- 86 Georgia Troullinou, Haridimos Kondylakis, Evangelia Daskalaki, and Dimitris Plexousakis. Ontology understanding without tears: The summarization approach. *Semant. Web*, 8(6):797–815, jan 2017. doi:10.3233/SW-170264.
- 87 Danai Vachtsevanou, Andrei Ciortea, Simon Mayer, and Jérémy Lemée. Signifiers as a first-class abstraction in hypermedia multi-agent systems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '23*, pages 1200–1208, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. doi:10.5555/3545946.3598763.
- 88 Danai Vachtsevanou, Philip Junker, Andrei Ciortea, Iori Mizutani, and Simon Mayer. Long-lived agents on the web: Continuous acquisition of behaviors in hypermedia environments. In *Companion Proceedings of the Web Conference 2020, WWW '20*, pages 185–189, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3366424.3383537.
- 89 Dylan Van Assche, Thomas Delva, Gerald Haesendonck, Pieter Heyvaert, Ben De Meester, and Anastasia Dimou. Declarative RDF graph generation from heterogeneous (semi-) structured data: A systematic literature review. *Journal of Web Semantics*, page 100753, 2022. doi:10.1016/J.WEBSEM.2022.100753.
- 90 Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014. doi:10.1145/2629489.
- 91 Antonia Wild, Andrei Ciortea, and Simon Mayer. Designing social machines for tackling online disinformation. In *Companion Proceedings of the Web Conference 2020, WWW '20*, pages 650–654, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3366424.3385770.