T2B2T: The Ontology for Adaptive Agent-Driven Seamless Integration with the Semantic Web

Carmelo Fabio Longo[1,*,†], Rocco Paolillo[2,†] and Miguel Ceriani[1,†]

[1]National Research Council, Institute of Cognitive Science and Technology, Via Giandomenico Romagnosi n. 18/A, Roma, 00196, Italy

[2]National Research Council, Institute for Research on Population and Social Policies, Via Palestro, 32, Roma, 00185, Italy

Abstract

Open science research, enabled primarily by shared Knowledge Graphs (KGs) whose backbone is the so-called Semantic Web, offers great opportunities to infer new information from prior-gathered data. Through the SPARQL language, it is possible to filter and process extracted data, from which implicit triples can be inferred, using reasoners such as Hermit, Pellet, etc. and also SWRL axioms. However, some KGs may lack the triples necessary for a specific research task and calculating these triples may require functions beyond SPARQL and OWL 2-based reasoners, which forces agents designers to delegate such task to high-level programming languages. In case of investigations by agent-based modeling involving the Semantic Web, these new triples must be integrated into KGs accordingly, otherwise a not seamless integration may invalidate aggregated outcomes given by interrelated inferences, in either mono- or multi-agent setup. In this paper we introduce the novel paradigm *Triples-to-Beliefs-to-Triples* (T2B2T) to model agent-based seamless integration with the Semantic Web, in order to adapt KGs and enable them to support Belief-Desire-Intention (BDI) inferences. Afterward, beliefs can be newly translated into triples, which will populate the derived KGs endowed with a legacy of past task-oriented inferences, achieved in the domain of BDI-logic.

Keywords

Semantic Web, BDI agents, Knowledge Graph, Multi-agent systems

1. Introduction

The concept of seamless integration with the Semantic Web [1] plays a fundamental role in scientific research applied to agents, whether they are virtual (as in social simulations), physical, or both (digital twins). In a FAIR perspective (Findable, Accessible, Interoperable, Reusable), each scenario can benefit from shared knowledge in the shape of triples, in order to either reuse data (virtual) or foster agent's interoperability (physical/digital twins), and the ways where access to such knowledge is more direct, by limiting abstraction layers, especially for real-time applications.

Proceedings of the Joint Ontology Workshops (JOWO) - Episode XI: The Sicilian Summer under the Etna, co-located with the 15th International Conference on Formal Ontology in Information Systems (FOIS 2025), September 8-9, 2025, Catania, Italy

*Corresponding author.

†These authors contributed equally.

fabio.longo@cnr.it (C. F. Longo); rocco.paolillo@cnr.it (R. Paolillo); miguel.ceriani@cnr.it (M. Ceriani)

0000-0002-2536-8659 (C. F. Longo); 0000-0001-9816-5839 (R. Paolillo); 0000-0002-5074-2112 (M. Ceriani)

In a broad sense, the focus on *agents* refers to the means by which an action is carried out. Through the application of specific constraints derived from both cognitive and computer science, the concept of "intelligent" agent emerged, together with "autonomous", i.e., capable of acting independently, exhibiting control over its internal state. M. Bratman [2] and his theory on human practical reasoning made a significant contribution to this, where the so-defined *mental attitudes*, namely *Belief*, *Desire* and *Intention* (BDI) represent respectively the information, motivational and deliberative states of the agent. Rao et al. [3] provided a more practical system for rational reasoning, which is a simplified version of the Procedural Reasoning System (PRS) [4, 5], one of the first implemented agent-oriented systems based on the BDI architecture, and a successor system, dMARS [6] (distributed Multi-Agent Reasoning System). However, they represent only beliefs about *current* state of the world (which can be expected to change over time), by considering only ground sets of literals with no disjunctions or implications, where the so-defined *plans* are being considered a special form of beliefs as means of achieving certain future *world* states. But when knowledge of such worlds, either current or future, is stored in shared Knowledge Graphs (KGs), the continuous access/inference/update on its content by agents can become cumbersome and potentially lead to inconsistencies. On the other hand, established agent engineering frameworks like JACK[1], JADE[2], JADEX [7] or JaCaMo [8] are not inherently designed to interact with the Semantic Web, nor do they offer an agent modeling language and environment that integrates seamlessly with them, since their introduction dates back a decade ago, when the importance of FAIR principles had not yet been sufficiently highlighted.
Another problem is that often KGs lack required triples for a research task, and these cannot be achieved with SPARQL language and OWL common reasoners (such as Hermit or Pellet), which forces delegating such operations to high-level programming language. The continuous updates of triples coming from such computation produce also an overhead that leads to a weak evaluation of the here-defined *aggregated* relations, that are a characterization of interrelated agents' inference outcomes, in either mono- or multi-agent setup. For instance, consider two agents acting on the same triples, either sequentially or in parallel (in the case of a multi-agent system). If the update process is delegated to a triple store management module, which operates independently of the inference engine, the results produced by the second agent may vary depending on the update timing of the triples coming from the first agent. For this reason, it is preferable to move the triples required for inference into the BDI agent's Knowledge Graph (KG), where appropriate measures can be applied to coordinate agents so to ensures consistent overall results. Subsequently, the triples can be updated either in the same KG or in a new one.

In light of the above, the main contribution of our work can be summarized as follows:

- We model the process of seamless integration of BDI agents and the Semantic Web, by introducing the novel paradigm *Triples-to-Beliefs-to-Triples* (T2B2T).
- We show that T2B2T paradigm paves the way to more sophisticated inferences involving computations out of SPARQL and OWL-based reasoners, in symbolic and sub-symbolic notation, shifting from open- to closed-world assumption.

[1] https://aosgrp.com.au/jack/
[2] https://jade.tilab.com/

- We show the effectiveness of the T2B2T and synchronization between agents through a formal exemplificative case study in Section 4.

2. Related Works

Some scholars have tried to integrate *multi-agent systems* (MAS) with ontologies. The authors of OASIS [9, 10] proposed an OWL-based agent model language, endowed with a fine-grained descriptions of the behavior of agents. However, their approach requires the executive grounding to be delegated to other frameworks. In this paper, we focus on the integration between Semantic Web resources and reactive reasoning on them. The authors of SW-CASPAR [11] leveraged Natural Language Processing (NLP) in a BDI framework, enabling meta-reasoning in the Semantic Web, albeit not providing templates to implement multi-agent coordination protocols and is not compliant to the well-known FIPA[3] agents interoperability guidelines. AJAN [12] uses a modular framework for building Semantic Web-enabled intelligent agents, built on Semantic Web standards and Behavior Tree technology. It supports SPARQL-extended behavior trees for agent scripting, multi-agent coordination, and is extensible with additional modules and communication layers. The advantage of using behavior trees over production rule systems, as we build on with the T2B2T paradigm, has not yet been documented. Production rule systems have historically been a foundational approach to artificial intelligence, such as in the General Problem Solver (GPS) [13]; on the other hand, behavior trees are primarily designed for applications in robotics [14], while they might exceed complexity in other fields. A special case of MAS is social simulation, where agents represent autonomous entities capable of processing information and interacting with their surrounding environment, modeling the foundations of collective behavior and changes in the system due to individual decisions. A more prevailing usage of ontologies in this field is to guarantee the interoperability of data either as input for models' initialization or as output for numerical analysis. For instance, the EPIK model [15] is built on a framework for epidemiological studies, initializing GIS-based simulations with relational datasets, to then use RDF and SQL formats to extract experiment results via query. The usage of semantic ontologies to model the cognitive architecture of agents and their execution of plans or communication seems underrated. Farrenkopf et al. [16] delved into this aspect. In their model for business decisions, they map ontologies into the cognition of agents mimicking BDI architecture. They sorted different layers of knowledge domain, such as an Abstract Domain Layer (ADL) and a Specific Domain Layer (SDL) related to the market sector, and an Individual Domain Layer (IDL) of local agents that evolved through experience and communication. By sharing information with others via communication, agents contribute to the update of the ADL and SDL layers. Our contribution with the implementation of T2B2T goes beyond these studies, using BDI to formalize the cognitive architecture of agents, plans of execution and beliefs updates, together with seamless integration between beliefs and semantic triples.

[3] http://www.fipa.org/

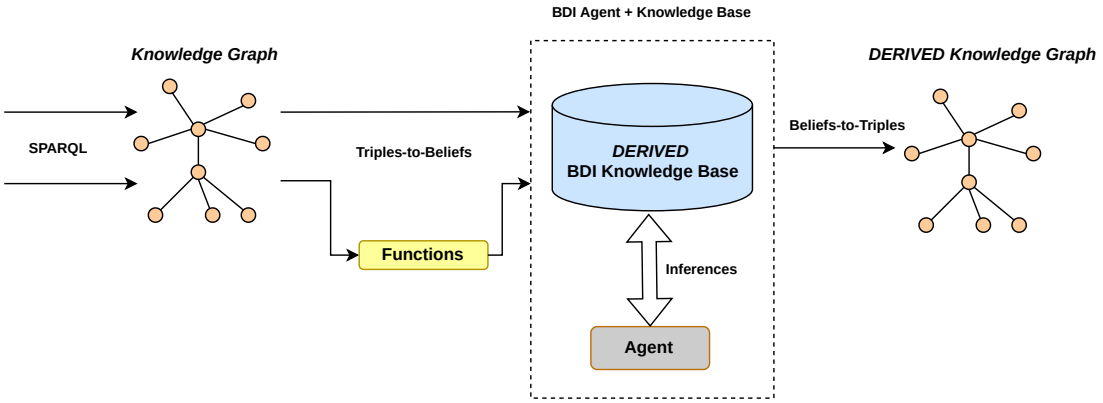

Figure 1: The simplified functional schema of the T2B2T paradigm

3. The T2B2T Ontology

The process we want to model in this contribution, i.e., the here-proposed *T2B2T* paradigm, is depicted in Figure 1. Let us suppose a starting KG from which one or more agents must make inference, and let us suppose that such KG lacks the required triples to carry out a specific task or searched information that require a process of inference or elaboration over such triples. Additional triples are being built starting from existent ones through external functions in high level language non computed by SPARQL. Afterward, all required triples are being translated in beliefs and asserted in the original knowledge base or a different one, realizing the start-pipeline of T2B2T. The updated knowledge graph can be used to support further inferences. Beliefs notation can span from symbolic to sub-symbolic, but in a deterministic setup we must focus on symbolic notation. In this configuration we can use Prolog-like engines to combine predicates with arity greater than two, overcoming the limitations of the SWRL [17] language, and leverage on inference criteria based on the backward-chaining algorithm rather than the forward-chaining of OWL-based reasoners like Pellet [18] or Hermit [19]. Inference must also be supported by an event queue, in order to coordinate agents and let them produce consistent outcomes in case of interrelated inferences, i.e, when outcome's inference (in terms of new triples) of an agent can affect inference used by another agent and subsequently their behavior. Figure 2 show three possible scenarios for two agents (AGT 1, AGT 2) interacting with the Semantic Web, running concurrently, each with its own cycle, which is being decomposed into three distinct stages: *Acquiring Triples Stage*, *Inference Stage* and *Updating Triples Stage*.

1. ***Acquiring Triples Stage* (ATS)**. In this stage, through SPARQL language, either local or remote, triples store are queried, to extract triples from one or more KGs (in case of federated[4] queries). Afterward, extracted triples are translated in beliefs and asserted in the agent's Knowledge Base (KB). With remote triple stores, the duration of this stage is affected by the congestion of both physical networks and remote machines hosting triples stores. Moreover, possible invocations of remote RESTful interface to compute the functions required for additional beliefs[5] assertions might also affect the duration of this stage wiht unpredictable timing.
2. ***Inference Stage* (IS)**. This stage's duration is affected by the computational resources of the machine hosting agents, which depends on both KB size and employed algorithms for inference.
3. ***Updating Triples Stage* (UTS)**. During this stage, triples resulting from the IS outcomes will be (possibly) copied to either origin KGs or other ones to be used for next inferences. This stage's duration, as well as ATS, can be affected by the congestion of both physical networks and remote machines hosting triples stores.

[4] https://www.w3.org/TR/sparql11-federated-query/
[5] Here defined as *Derived Beliefs*.

Figure 2 shows three possibile (non exhaustive) scenarios of interrelated inferences of the two agents.

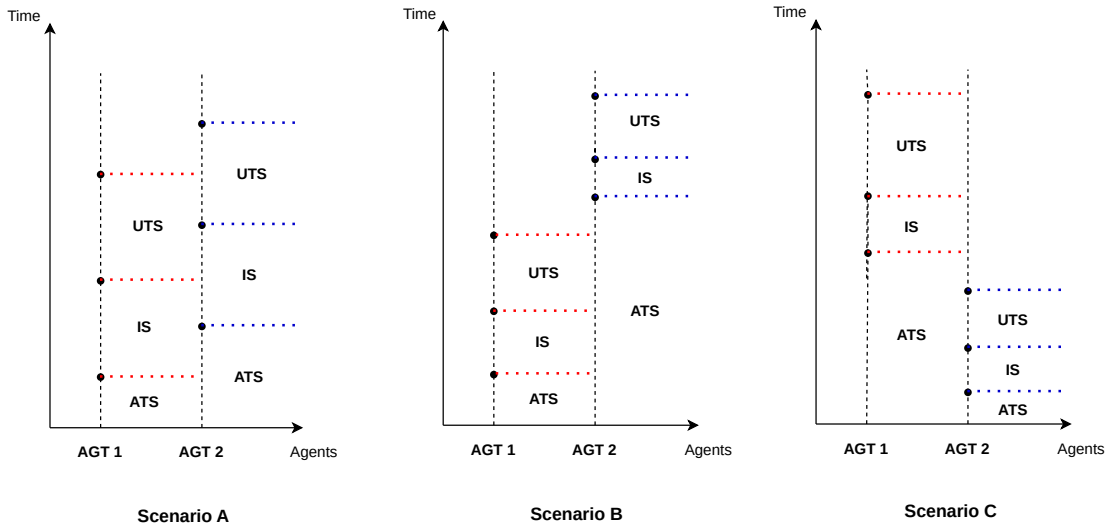

Figure 2: Three possible scenarios for two agents interacting with the Semantic Web, running concurrently, each with its own reasoning cycle. Legend: ATS, Acquiring Triple Stage; IS, Inference Stage; UTS, Updating Triple Stage

- **Scenario A**: in this scenario, the two agents start at the same time, but the ATS of AGT 2 is longer, taking more time than AGT 1 to extract triples from the triples store and populate its KB. Despite the delay, there are not interrelated phenomena between the two agents, since there is not overlapping between their ATSs and UTSs. So, their inferences are not mutually conditioned.
- **Scenario B**: in this case, there is clear overlapping between UTS and ATS of respectively AGT 1 and AGT 2, which means that acquired triples of AGT 2 will be affected by the parallel update of AGT 1.
- **Scenario C**: opposite to scenario B, due to the overlapping of ATS and UTS of respectively AGT 1 and AGT 2, the acquired triples of AGT 1 will be affected by the parallel update of AGT 2 in non-deterministic way, therefore interrelated inference will be also non-deterministic.

In the light of the above, in order to handle properly and study possible interrelated inference, the employment of a queue is mandatory, in order to regulate each stage execution according to wanted outcomes. The BDI frameworks Jason [20] and Phidias [21], for instance, based respectively on the Java and Python languages, are endowed with an event queue.

In the scope of this work *Desires* are not included in the ontology, due to the introduction of the so-called *Plans*, which are abstract specifications of both the means for achieving certain desires and the options available to the agent, therefore desires are implicitly described by

plans aimed to achieve the well-defined *Goal*. Moreover, following Rao et al. [3], each intention that the system forms by adopting certain plans of action is represented implicitly by using a conventional run-time stack of hierarchically related plans, that is why we omit to model them in the ontology. Here's some details of classes and properties modeling the T2B2T paradigm (Figure 3):

- ***Agent***. Instances of this class represent a single agent aimed to make inference on KGs.
- ***Belief***. Instances of this class are referenced by the object-property *hasBelief*, and refers to a piece of information that an agent considers to be true about the environment. Information can be (but not limited to) a predicate, having a label related to a property of a triple and two arguments (subject and object), and even a sub-symbolic representation in a vectorial space.
- ***Plan***. Instance of this class represent a set of intentions aimed at achieving a specific goal, and they are referenced by the object-property *hasPlan*.
- ***Action***. Instance of this class represent a primitive action or subgoal that has to be achieved for the plan's execution to be successful, and they are referenced by the object-property *hasAction*.
- ***Goal***. represents a desired state or outcome that the agent aims to achieve. Goals drive the agent's decision-making process and influence its intentions, and they are referenced by the object-property *hasGoal*.
- ***Triple***. Instances of this class represent triples from KGs, whose properties, subjects and objects are referenced by the data-properties *hasProperty*, *hasSubject*, *hasObject*, respectively.
- ***Enricher***. This is a subclass of *Action*, being part of a specific plan aimed to enriching agents KB with beliefs computed by a function referenced by the object-properties *hasFunction*. Among other *Action*'s instances related to *Goal*, before any inference by instances of *Agent*, an instance of the class *Enricher* represents the action of populating the KB with additional beliefs derived from triples, whom are not present in the origin KGs.
- ***DerivedBelief***. This is a subclass of *Belief*, whose instances represent a new belief not related to any triple present in the starting KG, which is a combination of existent beliefs and the function *Function* referenced by the object-property *hasFunction* of *Enricher*. The latter carry out the task of asserting the *DerivedBelief* in the agent's KB, which is referenced by the object-property *assertBelief*.
- ***Event***. Each instance of the class *Event* aims to model whatever kind of action aimed to change agent's KB content, and it is referenced by the object-property *hasEvent*. In the scope of this work, and event can be also sub-plans relate to starting/ending of ATS, IS and UTS (cf. Figure 2).
- ***Queue***. Instance of this class represents a queue to coordinate parallel agent's interaction with the KB, in multi-agent setting, in order to achieve deterministic outcomes. It is referenced by the object-property *isQueuedOn* of instances of the class *Event*.
- ***Query***. Instances of this class are about SPARQL queries that will feed agent's KB, after triples-to-beliefs translation, and they are referenced by the object-property *hasQuery*.

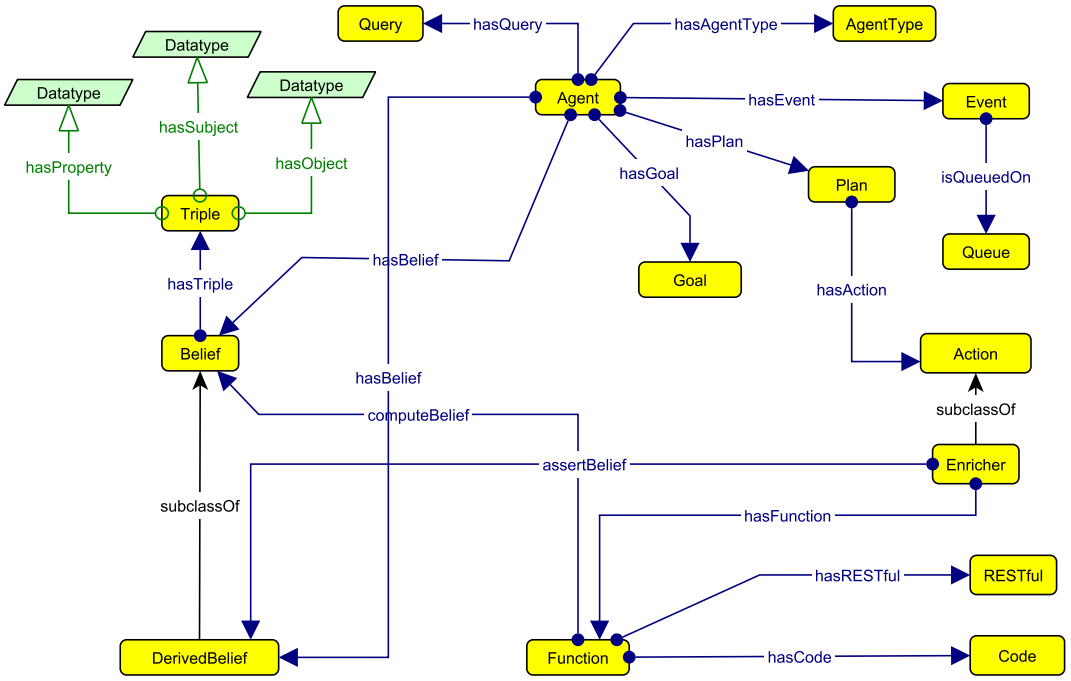

Figure 3: The ontology modeling the T2B2T paradigm

- ***AgentType***. Instances of this class represent entities whose behavior is simulated within the framework where inference takes place, and they are referenced by the object-property *hasAgentType*.
- ***Function***. Instances of this class represent functions computing newly-introduced beliefs, having (or not) in input beliefs translated from triples of the origin KGs, and they are referenced by the object-property *hasFunction*.
- ***RESTful***. Instances of this class represent remote RESTful interfaces to compute new *DerivedBelief*, and they are referenced by the object-property *hasRESTful*.
- ***Code***. Instances of this class represent code fragment to compute locally new *DerivedBelief*, and they are referenced by the object-property *hasCode*.

The T2B2T is implemented through the Semas framework (***SE**mantic **M**ulti-**A**gent **S**ystem*) built on the top of the Belief-Desires-Intention architecture. The BDI architecture in Semas is built on the top of Phidias [21], a Python tool for declarative programming with the ability to perform logic-based reasoning in Prolog style. Following the T2B2T paradigm as in Figure 1, triples are retrieved by Semas from a KG either through a local or a remote SPARQL query and turned into the beliefs system of an agent into their Knowledge Base (KB) used to generate inferences. The production rules for generation of plans out of inference of agents takes the form:

```
[TRIGGERING EVENT] / [CONDS] » [PLAN]
```

where the `[TRIGGERING EVENT]` placeholder refers to specific runtime events related to the agent, as the *Desire* of agents, which triggers the `[PLAN]` execution in case the presence of beliefs in KB is satisfied (`[CONDS]`). `[PLAN]` contains a list of *actions* which implicitly implement the designed *goal*, where each action can either execute code in high level language or assert/retract beliefs. At any time, the whole content of the KB can be newly translated (with specific built-in functions) in triples, either to update the origin KG or to build locally novel *derived* KGs, so to close the end-pipeline of T2B2T.

4. Case-Study

We provide a case-study to illustrate the three scenarios described in the previous section. We apply the T2B2T paradigm to a case of social simulation, where agents represent humans making decisions based on the inferences derived from their own KB which can share beliefs or not with KB of other agents. This section wants to show how the parallel activation of *ATS, IS, UTS* stages of agents and their influence can affect the aggregated outcomes derived from the execution of plans based on inferences. To this aim, we build a formal model of academic mobility as shown in the baseline of Figure 4. Twelve agents, including AGT 1 and AGT 2 are affiliated with 4 universities (Uni1, Uni2, Uni3, Uni4). AGT 1 at Uni1 has as a preference for topic *red*, whose top-author is scholar AGT2b at Uni2. The elective field of AGT 2 at Uni2 is *green* topic, whose top-author is AGT1b at Uni1. Both AGT 2 and AGT 1 have been offered a position at Uni3 and Uni4 and they have to decide whose offer to accept. The production rules to make a decision are based on the assumption that scholars aim at connecting with top-authors in their elective field [22] and small-world network mechanisms [23]. In no one of alternative universities a top-author in the elective field is available for either of the two agents. Thus, they have to select the university which hosts co-authors to the top-authors of interest, who might potentially connect with them. We report a multi-agent version of SEMAS, with *main* agent collecting changes in the model due to the decisions of agents AGT 1 and AGT 2.

Figure 5 shows the three stages of the T2B2T paradigm applied to the case. In line 4, the *Acquiring Triples Stage (ATS)* is launched by the procedure `load()`, which allows agents to extract a KB from triples through a local SPARQL request, in this case. In the *Inference Stage (IS)* in line 7, the production rule as described in section 3 is triggered starting from `DesireGoalFor(X,D,U)` for field X and the choice between universities D and U scholar S has been selected for (placeholders for Uni3 and Uni4). The reasoner scans for what co-author Z to top-author Y in the field X exists and affiliated to which university. In the example, university U matches the condition. As such, the inference suggested `+AcceptOffer()` triggers the *Updating Triples Stage (UTS)* in the line 11, sending the communication of the new affiliation to university U of agent S to the *main* agent through `send()`. This procedure specifies *main* as the receiver A of the communication, who receives in line 19 a new `+TRIPLE(S,H,L)` to assert, where S scholar is the subject of the triple, H its predicate (`"hasAffiliationWith"`), and L its object (university U). The update of the system implies the deletion of all previous selectionship status associated to agent S (`-Selectionship(S,P)`) and its previous affiliations (`-Affiliation(S,P)`), where P is a placeholder to any object of the beliefs retracted by the `UpdateMain()` procedure.

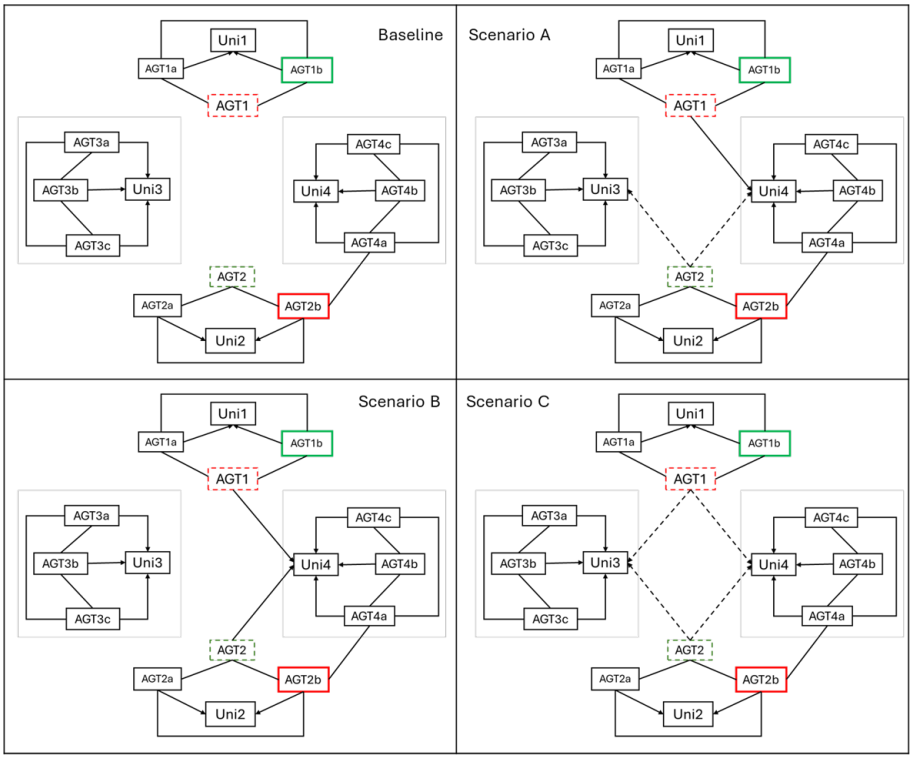

Figure 4: Comparison of three scenarios: initialization and synchronization. Solid lines denote deterministic outcomes, dashed lines not deterministic outcomes inferable at time 0

Figure 4 shows how the selection of agents and the final configuration of the system can change due to the different scenarios of synchronization described in section 3. As a measure of aggregated outcome, we report the affluence of universities measured with a *centrality* index, computed as the number of affiliations to that university divided by the number of scholars. In the baseline scenario, where each university hosts three scholars, each university has an index equal to 0.25, which will change as effect of the decision of AGT 1 and AGT 2. The scenario *A* implies a condition of semi-uncertainty. The ATS stage of AGT 1 starts first and independent on AGT 2. The inference process will select Uni4 deterministically, due to the chance to connect with top-author AGT2b at Uni2 through co-author AGT4a. AGT 2 has no knowledge of what is the choice of AGT 1. Even though Uni3 would be the best choice due to the affiliation of AGT 1 who could connect with top-author AGT1b, this information would be excluded from the IS stage of AGT 2. So, both Uni3 and Uni4 would have equal chance to be selected by the agent, not providing any of them an actual advantage. In both cases, Uni4 would increase its affluence with 1 new affiliation (AGT 1), while there is uncertainty for Uni3 or Uni4. The Scenario *B* is deterministic instead, since the initial conditions at the baseline and the synchronization pattern could only lead to one solution. In this scenario, AGT 1 starts first, communicating its changes before AGT 2 acquires the necessary triples. AGT 1 will select Uni4, now known to AGT 2. The agent can thus make the inference that AGT 1, now affiliated with Uni4 could introduce to the topauthor AGT1b. Uni4 would be preferred to Uni3, so that the affluence index can be

```
## Local agent execution (AGT1, AGT2)

# Acquiring Triples Stage (ATS)
load() >> [show_line("\nAsserting all OWL 2 beliefs triples...\n"), assert_beliefs_triples(),
     pre_process()]

# Inference Stage (IS)
DesireGoalFor(X,D,U) / (Selectionship(S,D) & Selectionship(S,U) & TopAuthorship(Y,X) &
CoAuthorship(Z,Y) & Affiliation(Z,U)) >> [-CoAuthorship(Z,Y), +AcceptOffer(U)]

# Updating Triples Stage (UTS)
+AcceptOffer(U) >> [+Affiliation(U), send("main","hasAffiliationWith",U)]

# Communication to Main agent
send(A,H,L) >> [ +AGT(A), +COMMUNICATE(H,L)]
+COMMUNICATE(H,L) / AGT(A) >> [-AGT(A), +COMMUNICATE(H,L)[{'to': A}]]

## Main agent execution

+COMMUNICATE(H,L)[{'from': S}] >> [show_line("received belief from ", S), UpdateMain(S),
+TRIPLE(S,H,L), pre_process()
UpdateMain(S) / Selectionship(S,P)  >> [-Selectionship(S,P), UpdateMain(S) ]
UpdateMain(S) / Affiliation(S,P)  >> [-Affiliation(S,P), UpdateMain(S)]]
```

Figure 5: Case study for scenarios of synchronization

deterministically computed at time 0: higher centrality for Uni4 with 2 new affiliations for a total of 5 affiliations (score 0.42), Uni3 with 3 affiliations (score 0.25), Uni1 and Uni2 decreasing to 2 affiliations each (score 0.17). The scenario *C* leads to a condition of uncertainty. In this case, AGT 2 is the first agent to acquire triples and make inference, with no alternative offering an advantage compared to the other, so to choose randomly. The choice of AGT 1 will depend on the selection of AGT 2, as couathor to AGT2b, which remains unknown at time 0. As such, is not possible to infer a final outcome at this step.

5. Conclusions

In this paper, we devise a paradigm as guideline for a seamless integration of multi-agent systems (MAS) with the Semantic Web, proposing the novel paradigm *Triples-to-Beliefs-to-Triples* (T2B2T) ontologically formalized within a *Beliefs-Desires-Intentions* (BDI) framework and implemented with SEMAS architecture. The T2B2T paradigm allows agents to reason upon RDF KGs, deciding on which action to take and updating internal beliefs accordingly, propagating new inferences back into the Semantic Web ecosystem as new formed triples they can communicate. We showed an application of the paradigm with a formal model and how different sequences of parallel reasoning of agents and synchronization can lead to different aggregated outcomes due to the limited portion of knowledge the individual agent elaborates. T2B2T paradigm can enhance interoperability of data for seamless integration and reasoning formalization in multi-agent systems. Further studies can explore the potentiality of the tool for the connotation of agents in MAS application both for scholarly inquiry as we did with this formal study and machine coordination as in Internet of Things (IoT) and Web of Things (WoT) implementations.

Acknowledgments

This work was supported by FOSSR (Fostering Open Science in Social Science Research), funded by the European Union – NextGenerationEU under NRRP Grant agreement n. MUR IR0000008.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] T. Berners-Lee, J. Hendler, O. Lassila, Web Semantic, Scientific American 284 (2001) 34–43.
[2] M. Bratman, Intention, Plans, and Practical Reason, Harvard University Press, 1987.
[3] A. Rao, M. Georgeff, BDI agents: From theory to practice, in: Proceedings of the First International Conference on Multi-agent Systems (ICMAS-95), San Francisco, CA, 1995, pp. 312–319.
[4] M. Georgeff, A. Lansky, Procedural knowledge, Proceedings of the IEEE 74 (1986) 1383–1398. doi:10.1109/PROC.1986.13639.
[5] F. Ingrand, M. Georgeff, A. Rao, An architecture for real-time reasoning and system control, IEEE Expert 7 (1992) 34–44. doi:10.1109/64.180407.
[6] M. D'Inverno, M. Luck, M. Georgeff, D. Kinny, M. Wooldridge, The dMARS architecture: A specification of the distributed multi-agent reasoning system, Autonomous Agents and Multi-Agent Systems 9 (2004) 5–53. URL: https://doi.org/10.1023/B:AGNT.0000019688.11109.19. doi:10.1023/B:AGNT.0000019688.11109.19.
[7] L. Braubach, A. Pokahr, W. Lamersdorf, Jadex: A BDI-agent system combining middleware and reasoning, in: R. Unland, M. Calisti, M. Klusch (Eds.), Software Agent-Based Applications, Platforms and Development Kits, Birkhäuser Basel, Basel, 2005, pp. 143–168.
[8] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, A. Santi, Multi-agent oriented programming with JaCaMo, Science of Computer Programming 78 (2013) 747–761. URL: https://www.sciencedirect.com/science/article/pii/S016764231100181X. doi:https://doi.org/10.1016/j.scico.2011.10.004, special section: The Programming Languages track at the 26th ACM Symposium on Applied Computing (SAC 2011) & Special section on Agent-oriented Design Methods and Programming Techniques for Distributed Computing in Dynamic and Complex Environments.
[9] D. Cantone, C. F. Longo, M. Nicolosi-Asmundo, D. Santamaria, C. Santoro, Towards an Ontology-Based Framework for a Behavior-Oriented Integration of the IoT, in: Proceedings of the 20th Workshop From Objects to Agents, 26-28 June, 2019, Parma, Italy, CEUR Workshop Proceeding Vol. 2404, 2019, pp. 119–126.
[10] D. Cantone, C. F. Longo, M. Nicolosi Asmundo, D. F. Santamaria, C. Santoro, Ontological smart contracts in oasis: Ontology for agents, systems, and integration of services, in: D. Camacho, D. Rosaci, G. M. L. Sarné, M. Versaci (Eds.), Intelligent Distributed Computing XIV, Springer International Publishing, Cham, 2022, pp. 237–247.
[11] C. F. Longo, C. Santoro, M. Nicolosi-Asmundo, D. Cantone, D. F. Santamaria, Towards

ontological interoperability of cognitive IoT agents based on natural language processing, Intelligenza Artificiale 16 (2022) 93–112. doi:10.3233/IA-210125.
[12] A. Antakli, A. Kazimov, D. Spieldenner, G. E. J. Rojas, I. Zinnikus, M. Klusch, Ajan: An engineering framework for semantic web-enabled agents and multi-agent systems, in: P. Mathieu, F. Dignum, P. Novais, F. De la Prieta (Eds.), Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection, Springer Nature Switzerland, Cham, 2023, pp. 15–27.
[13] A. Newell, H. A. Simon, GPS, a program that simulates human thought, in: H. Billing (Ed.), Lernen und automatische Informationsverarbeitung, Springer, 1961, pp. 109–124.
[14] P. Ögren, C. I. Sprague, Behavior trees in robot control systems, Annual Review of Control, Robotics, and Autonomous Systems 5 (2022) 81–107. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-042920-095314. doi:https://doi.org/10.1146/annurev-control-042920-095314.
[15] S. S. Hasan, E. A. Fox, K. Bisset, M. V. Marathe, EpiK: A knowledge base for epidemiological modeling and analytics of infectious diseases, Journal of Healthcare Informatics Research 1 (2017) 260–303. doi:10.1007/s41666-017-0010-9.
[16] T. Farrenkopf, M. Guckert, N. Urquhart, S. Wells, Ontology based business simulations, Journal of Artificial Societies and Social Simulation 19 (2016). doi:10.18564/jasss.3266.
[17] World Wide Web Consortium, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004. URL: http://www.w3.org/Submission/SWRL/.
[18] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, Y. Katz, Pellet: A practical OWL-DL reasoner, Web Semantics 5 (2007) 51–53.
[19] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, Z. Wang, HermiT: An OWL 2 Reasoner, Journal of Automated Reasoning 53 (2014) 245–269.
[20] R. H. Bordini, J. F. Hübner, M. Wooldridge, Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology), John Wiley & Sons, Inc., Hoboken, NJ, USA, 2007.
[21] F. D'Urso, C. F. Longo, C. Santoro, Programming intelligent IoT systems with a Python-based declarative tool, in: The Workshops of the 18th International Conference of the Italian Association for Artificial Intelligence, 2019.
[22] V. A. Hill, Collaboration in an academic setting: Does the network structure matter?, Center for the Computational Analysis of Social and Organizational Systems (2008).
[23] M. A. Koseoglu, Growth and structure of authorship and co-authorship network in the strategic management realm: Evidence from the strategic management journal, BRQ Business Research Quarterly 19 (2016) 153–170. doi:10.1016/j.brq.2016.02.001.

A. Online Resources

The sources for the code presented are available via

- Semas,
- Case study.