Geometric and Spatial Reasoning in BDI Agents: a Survey

Angelo Ferrando[1,*,†], Andrea Gatti[2,*,†] and Viviana Mascardi[2,*,†]

[1]*University of Modena-Reggio Emilia, Italy*

[2]*University of Genova, Italy*

Abstract

In this paper we survey the literature on geometric and spatial modelling and reasoning in Belief-Desire-Intention (BDI) agents and in Logic-Programming based approaches. The motivation for this survey is the VEsNA framework, where cognitive agents implemented using the Jason BDI language and immersed in an unknown virtual environment must build a map of that environment and reason about it.

Keywords

Literature Review, Geometric Modelling, Spatial Reasoning, Belief-Desire-Intention, Logic Programming, Prolog, Virtual Reality, VEsNA

1. Introduction

According to the well known definition by Jennings, Sycara and Wooldridge [1], an intelligent agent is a computer system, situated in some environment, capable of autonomous, social, reactive, and pro-active actions. The strong definition of agents adds to these characteristics the requirement of being conceptualised in terms of mentalistic features like beliefs, desires, goals, intentions, leading to the notion of cognitive agents. The most known architecture for cognitive agents is the Belief-Desire-Intention (BDI) one [2].

We believe that, in the ChatGPT [3] and metaverse [4] era, agent's sociality should involve humans by design, and situatedness should take virtual reality into account. Hence, our vision of the next generation intelligent software agents involves cognition and reasoning abilities, natural language interaction, and virtual reality.

To make our vision concrete we designed and developed VEsNA [5], a general-purpose and agent-based framework for managing *V*irtual *E*nvironments via *N*atural language *A*gents. VEsNA is freely available to the research community from https://github.com/driacats/VEsNA, and seamlessly integrates cognition, natural language, and virtual reality. Currently, we are addressing the challenging research questions

- **RQ1**: *how can VEsNA agents perceive the unknown virtual reality they are situated in?*,

- **RQ2**: *how can they create and maintain a geometric representation of their environment?*,
- **RQ3**: *how can they reason about spatial relations?*.

Although preliminary and subject to improvements, we have an answer for **RQ1**. This paper paves the way to answering questions **RQ2** and **RQ3** by presenting the results of a review carried out on the relevant related literature. We do not aim at answering the two questions above in this paper; rather, we aim at identifying the works where answers can be found.

The paper is organized in the following way. Section 2 provides the context for our research by introducing VEsNA. Section 3 provides pointers to recent papers useful for answering **RQ2**, while Section 4 presents the result of our Systematic Literature Review (SLR) on BDI agents and spatial reasoning. To the best of our knowledge no such review, providing a partial answer to **RQ3**, exists in the recent literature; for this reason, we addresses the analysis of the literature in a systematic and reproducible way. Finally, Section 5 concludes.
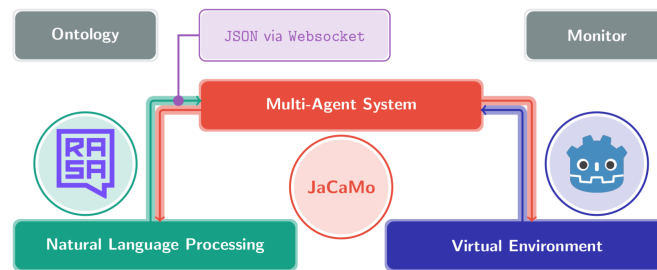
2. VEsNA

Figure 1: VEsNA architecture.

VEsNA is a general-purpose and agent-based framework for managing Virtual Environments via Natural language Agents. As shown in Figure 2, VEsNA integrates

 (i) a chatbot-like interface to support the sociality of agents; thanks to the chatbot supporting natural language interaction, sociality is not limited to agent-agent conversations, but it is extended to agent-human ones;

 (ii) a framework for building the dynamic virtual environment; virtual reality is very suitable for supporting agents' situatendness: therein, agents are expected to perceive the reality and to act over it;

 (iii) a framework for implementing cognitive, declarative agents able to reason about knowledge (about what users say, about the environment, about themselves) and to provide human-readable explanations; the cognitive framework supports the agents' autonomy, reactivity, proactivity.

In VEsNA's first release, the chatbot was implemented in Dialogflow [6] and the virtual environment in Unity [7]. In the current release, we substituted Dialogflow with Rasa [8, 9], which is an open source project, and Unity with Godot [10], again fully open source and

lighter than Unity. In both releases the framework for cognitive agents was JaCaMo [11]. For the implementation of BDI agents, JaCaMo exploits the Jason interpreter for the AgentSpeak declarative language [12].

In our recent research, we addressed a few key questions related with the two boxes depicted in gray in Figure 2, namely how to integrate and exploit ontologies in VEsNA, and how to monitor interactions among agents and between agents and humans.

In [13] we described On2Conv that takes in input an ontology modelling the domain of interest and generates both the skeleton for the Jason agent's plans and the intents for the Dialogflow chatbot, coherently with the provided ontological knowledge. On2Conv ensures that the chatbot and the Jason agents in charge for managing messages from the chatbot are aligned, since they are automatically created from the same ontology. The On2Conv extension, meant at using ontologies at design time to drive the generation of code, is not available in the VEsNA current release since it targets Dialogflow as the chatbot. We are currently porting it to Rasa.

In [14], instead, we proposed RV4Rasa, a Runtime Verification framework that monitors whether a given chatbot deviates from its expected behaviour. Its instantiation to monitor Rasa chatbots is presented, and the integration in VEsNA is available.
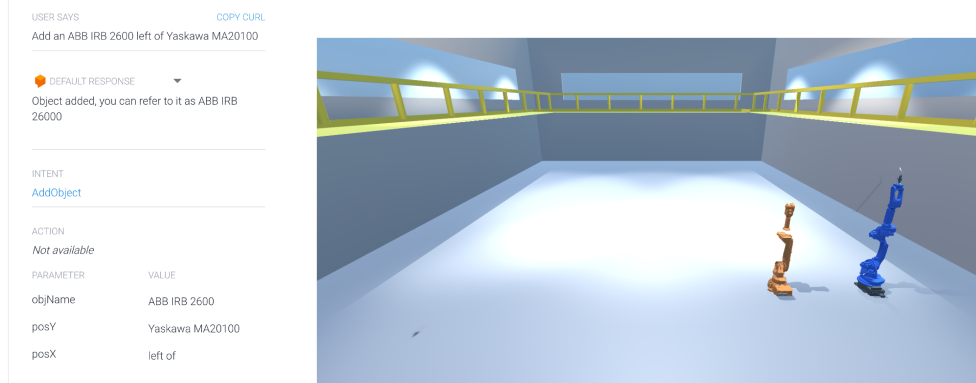
(**a**) The Dialogflow chat.   (**b**) The graphical result.

Figure 2: Interacting with Unity through Dialogflow thanks to VEsNA.

The screenshot[1] in Figure 2 shows a chatbot interface on the left, where the human user asks to add an ABB IRB 2600 industrial robot on the left of a previously placed robot, a Yaskawa MA 20100. The request is parsed by the chatbot, is then passed to the cognitive agent coordinating the VEsNA framework, and the agent creates and positions the new robot - that is just an object - in the virtual environment shown on the right. Note that no autonomous intelligent agent is inserted yet in that environment.

The challenge that we are facing now, is allowing VEsNA autonomous agents to be born in the virtual reality, to create a map of the unknown, dynamic environment they are situated in, and to behave accordingly based on some long term goals, specified in a declarative way.

---

[1]The screenshot is taken from the first VEsNA release where we used Dialogflow and Unity.

```
①  var objects = sight.get_overlapping_bodies()
```

objects

② Does r_1 from a to o_1 intersect other objects?   *No*
   Does r_2 from a to o_2 intersect other objects?   *Yes, o_1*
   Only o_1 is visible to the agent a, o_2 is occluded, sending:
```
      {perception: sight,
        data: {object: o₁, rotation: up, distance: near}}
```

③ The message is received by the environment artifact and a signal is sent to the agent:
```
   signal("seen", parseLiteral(obj), parseLiteral(rot), parseLiteral(dist);
```

④ The actor reacts to the signal, knows its actual position and adds a belief:
```
      +seen(Object, Rotation, Distance)
         :    position(X, Y)
         <-   +saw(X, Y, Direction, Object, Distance).
```
   Assuming the agent in position $(0, 0)$ the new belief is:
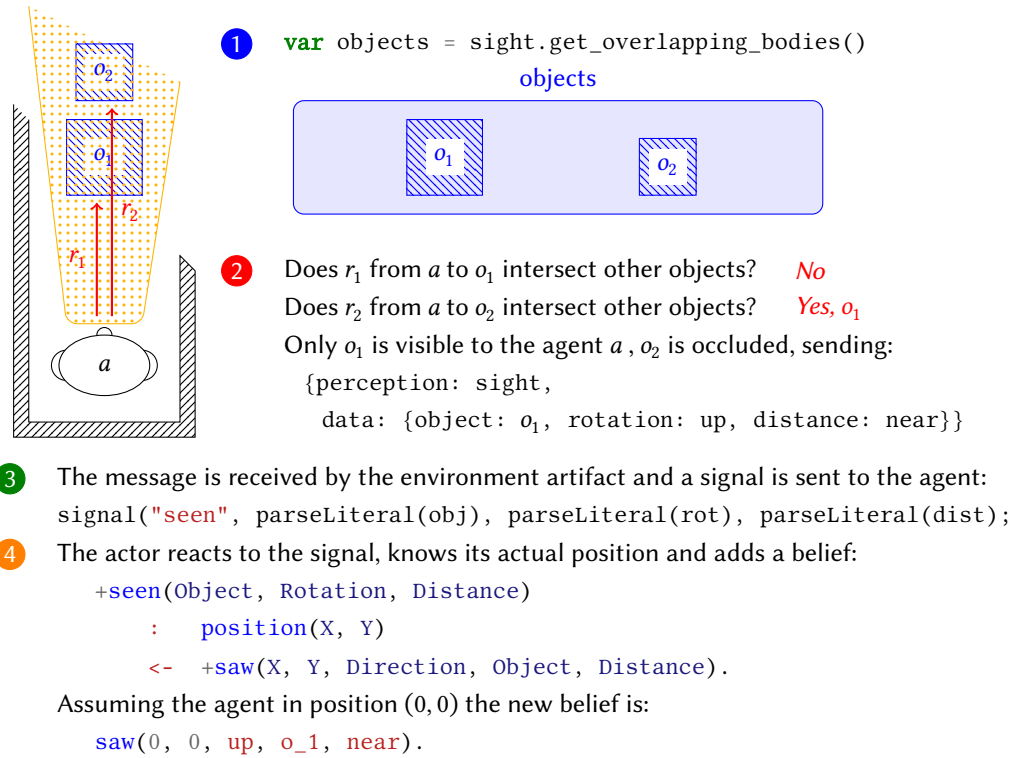```
      saw(0, 0, up, o_1, near).
```

**Figure 3:** VEsNA's answer to **RQ1**.

This requires to create a connection between the brain of the agents, implemented in Jason, and their virtual body implemented in Godot. The connection must allow the agents to perceive the environment and create, and timely update, a mental map of it.

The implementation of the way VEsNA agents see the environment and port what they see into their brain consists of two interconnected components:

1. The physical view implemented on the Godot side;
2. The transition from perception to belief on the JaCaMo side.

As shown in Figure 2 the connection between the two is implemented using a WebSocket connection. The agent instantiated in the virtual scene is equipped with a cylindrical collider, namely *viewcone* similar to the one described by Aversa et al. [15], that moves by following the head and intercepting objects that are then in front of the agent. With this first control, however, the viewcone intercepts all objects in an area without considering possible occlusions that make the object not visible to the agent. In order to take also occlusions into account, we added a further raycast step: we create a vector from the agent's head to the seen object and check that there are indeed no other objects between the two. In this case Godot sends a message to JaCaMo containing the name of the seen object, the rotation of the agent at the moment of perception and the distance at which it is located.

A JaCaMo environment artifact receives the message and sends a signal to the agent with
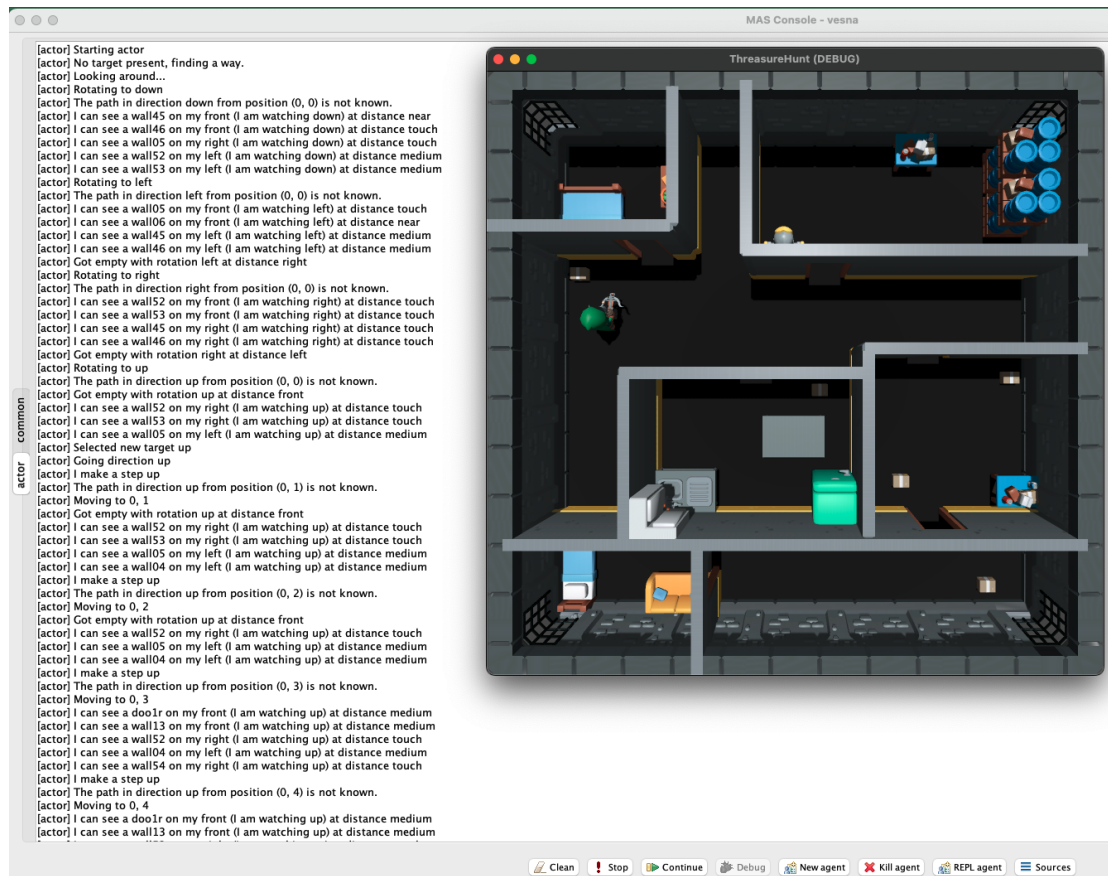
**Figure 4:** A VEsNA agent starting to explore its virtual environment.

the information. The agent intercepts the signal and creates a new belief where it stores the position, rotation, distance and object seen.

In Figure 2 we show the results of our very preliminary experiments, where one agent sees a wall and incorporates the belief that a wall is there, into its belief base.

This technical solution represents our current answer to **RQ1**. While we are able to represent beliefs about the surrounding virtual environment, the representation is too limited and ad-hoc to scale up to more realistic scenarios, and to be extended with meaningful semantic information. This is where the review presented in the next sections to answer **RQ2** and **RQ3** comes into play.

## 3. RQ2: Geometric Modelling in BDI agents

To review the literature on geometric modelling in BDI agents, we exploited Google Scholar setting 2005–2024 as time range and relevance as ranking criterion. The query we submitted to Google Scholar, (`"geometry"` OR `"geometric modeling"` OR `"geometric modelling"`)

`AND` `"BDI"`, allowed us to expand our horizons far beyond computer science. We learned that BDI may mean Boron–Diindomethene, which has interesting applications in bioanalytical processes, but also Beck Depression Inventory and Blood Damage Index. In the domain of insulators and superconductors, BDI is a chiral orthogonal class, whatever this means! Out of the first 20 papers returned by the query above, only two used the acronym BDI as Belief-Desire-Intention.

The first, by Antunes et al. [16], has nothing to do with geometric modelling; it deals with the origin, adoption and evolution of desires in BDI architectures, and was returned because of the title, "The Geometry of Desire". The second, by YenChern et al. [17], describes a very simple exercise of modelling a fire evacuation scenario using well known agent-oriented methodologies. The authors claim that they implemented the scenario in Unity 3D using a BDI plugin, which would be very interesting for our research, but the plugin is not explained and referenced in the paper. Also this second paper does not answer **RQ2** in any way.

The query (`"geometry"` `OR` `"geometric modeling"` `OR` `"geometric modelling"`) `AND` `"Belief Desire Intention"` returned results related with the BDI architecture as we intend it, but still none provided us with useful hints on how to model the geometric features of a 3D environment in a generic BDI framework. Many of the returned papers [18, 19] deal with integrating the BDI architecture with GAMA, an environment for implementing and running "spatially explicit" multi-agent simulations [20]: they include the word geometry for this reason, but the GAMA approach to support BDI agents cannot be easily ported to Jason.

In the Jason framework, beliefs are modelled as Prolog facts. We hence extended the query to (`"geometry"` `OR` `"geometric modeling"` `OR` `"geometric modelling"`) `AND` `"prolog"`. This allowed us to retrieve a very recent paper [21] that overviews some notable Prolog tools for automated reasoning in geometry, and provides details on ArgoTriCS, the Automated Triangle Construction Solver [22].

We stopped our search here, as the survey included in [21] represents a good starting point for answering **RQ2**, and our goal at this stage was to identify pointers to the relevant literature, not the final answer.

Our impression however is that, even if we had continued our search and carried out a more systematic review, the results would have been very limited. Integration of geometric modelling features inside BDI agents has never been explored as a research problem on its own, but always as parts of projects and experiments, leading to ad-hoc solutions. The study of the connections between computational geometry and Prolog raised interest in the eighties and nineties of the last century [23, 24, 25, 26], but it does not seem to be very fashionable nowadays.

For these reasons, we preferred to devote more time and effort to surveying works related with spatial reasoning, as the results we obtained – presented in Section 4 – were more abundant and promising.

## 4. RQ3: Spatial Reasoning in BDI Agents

The review of the literature on spatial reasoning in BDI agents was carried out in a systematic way. We issued the four queries `"spatial reasoning"` `AND` `"BDI"`, `"spatial reasoning"` `AND` `"belief desire intention"`, `"spatial reasoning"` `AND` `"prolog"`,

| Ref. | Year | Model. lang. | Impl. lang. | Other lang. |
|---|---|---|---|---|
| [27] | 2008 | None | Unclear | NetLogo+BDI+FIPA-ACL |
| [28] | 2010 | LP | AgentSpeak | Jadex, JTS Topol., PostGIS |
| [29, 30, 31] | 2011-2015 | CLP | CLP(QS) (SWI-Prolog) | C++, REDUCE |
| [32] | 2011 | Modal Logic | Prolog-like | NetLogo+BDI |
| [33] | 2012 | ASP | GQR | Perl, C++, Gringo |
| [34, 35] | 2015-2017 | ASPMT | ASPMT(QS) | None |
| [36] | 2015 | LP | Unclear | Unclear |
| [37] | 2016 | CLP | CLP(QS) (SWI-Prolog) | FreeCAD CGCS |
| [38, 39, 40] | 2016-2018 | LP | Prolog-like | NetLogo+BDI |
| [41] | 2017 | LP | Prolog | None |
| [42] | 2017 | LP | SWI-Prolog | C++ |
| [43] | 2018 | ASP | Clingo 5.2.0 | None |
| [44] | 2018 | LP | LogC-QSD (SWI-Prolog) | None |
| [45] | 2018 | LP | Prolog | None |
| [46] | 2021 | LP | SWI-Prolog | Arcmap |
| [47] | 2021 | GeoJSON | GeoJSON + 2P-Kt Prolog | Tile38, Lattuce |
| [48] | 2023 | LP, CLP | SWI-Prolog | OpenCV, Tesseract OCR |

**Table 1**

High level formalism for modelling spatial concepts (**Model. lang.**), language for implementing spatial concepts (**Impl. lang.**), other languages and frameworks used for the implementation (**Other lang.**).

"spatial reasoning" AND "logic programming" to Google Scholar, setting the last 20 years as range for the results, and ranking by relevance. For each query, we analysed the first 10 results. We excluded books, technical reports, PhD theses, master theses (apart one that was returned by multiple queries and was hence included because of its relevance) and papers already retrieved with previous queries. We also excluded papers that presented methodological approaches, with no implementation, unless they were part of a research stream by the same group. After these search and filtering stages, 22 papers entered our review. They appear, listed in chronological order, in Tables 1 and 2. Papers belonging to the same stream of research are listed together in the same row.

Given that our goal is to equip our Jason agents with geometric and spatial reasoning capabilities, the more relevant papers are those where BDI appears in column **Agnt**, in Table 2.

Although returned by our search, the work by Sakellariou et al. [27] deals with spatial reasoning is a very marginal way. It describes the integration of BDI mechanisms in the agent-based modelling and simulation tool NetLogo [54]. The integration is not available to the community, and the notion of space is important only because it is important – by design – in NetLogo, where turtles, namely mobile agents, are expected to move around to accomplish their tasks. In NetLogo turtles are situated in a discrete torus-like 2D environment (with 3D extensions not mentioned in the paper) and can perceive features of the squares (patches) they are positioned in, and those around. In [27] spatial information is modelled as it is normally modelled in NetLogo, with assertions like "fire-at [23 15]". No spatial calculus is adopted, and no support to temporal reasoning is provided. The case study is a simulation of

| Ref. | T. | Calculus | Agnt | Case study | A. |
|---|---|---|---|---|---|
| [27] | No | None | BDI | Forest fire scenario | No |
| [28] | No | RCC | BDI | Autonomous, collision-free motion in dynamic environments in a logistic domain | No |
| [29], [30], [31] | No | 2D point-based spatial calculi | No | Geometric reasoning from Computer-Aided Architecture Design | No |
| [32] | Yes | Behind, FrontOf, RightOf, LeftOf, On | BDI | Simple example from traffic domain | No |
| [33] | Yes | QCN | No | Temporal reasoning formalisms, the Interval Algebra and RCC | No |
| [34], [35] | Yes | Non-monotonic spatial reasoning | No | Ramification problem; geometric reasoning; frame problem; abductive reasoning | [49] |
| [36] | Yes | QTC, QMPC, RCC | BDI | Fleet of taxi scenario | No |
| [37] | No | RCC | No | Circle contact; spiral chain; lamp layout | No |
| [38], [39], [40] | Yes | RCC | BDI | Simple example with agents using electric cars navigating to public charging point | [50] |
| [41] | No | RCC | No | Simple examples with manually encoded data representing geographical data | No |
| [42] | No | None | Robot swarm | Simulated & implemented experiments with multi-robot swarms | No |
| [43] | Yes | TC-6 and TC-10 | No | Fleet of taxi, T-Drive dataset | [51] |
| [44] | No | QSD, QSD-Jux, PLCA | No | Spatial tests on the LogC-QSD-Dataset | No |
| [45] | No | RCC | No | Layout of Multimedia documents | No |
| [46] | No | Entity- and Relation-Based Programming | No | Analysis of accidents in Berlin | [52] |
| [47] | No | Nearby, within, intersect | No | None | No |
| [48] | No | None | No | Maths puzzles assigned to children of 4th and 5th grade of primary school | [53] |

**Table 2**
Support to temporal relations (**T.**), principal calculus used for spatial reasoning (**Calculus**), integration of the spatial reasoning capabilities into intelligent agents or robots (**Agnt.**), **Case study**, availability of the implemented reasoner/framework to the research community (**A.**).

a multi-agent system in a simple scenario that involves forest fires detection and suppression.

The work by [28] is inspired by the region connection calculus (RCC) for qualitative spatial reasoning [55]. In particular, it uses RCC8 that consists of 8 basic binary relations possible between two regions: disconnected (DC), externally connected (EC), equal (EQ), partially overlapping (PO), tangential proper part (TPP), tangential proper part inverse (TPPi), non-tangential proper part (NTPP), non-tangential proper part inverse (NTPPi). They define beliefs and plans using the Jason syntax. As an example of plan exploiting spatial relations and reasoning upon them, the `avoid_collision` plan looks like

```
+!avoid_collision(PA, A): true <-
detect_spatial_relations_for_movement(ID, PA, RMR, MA);
```

```
detect_spatial_relations_for_obstacle(ID, PA, ROR, OA);
reason_spatial_composition(ID, RMR, ROR,PA, MA, OA, ...);
-+collision(CA);
!move_to_position(PA, MA, OA, A).
```

Even without going into the technical details of the plan syntax, where some variables have been omitted for readability, it is clear from the names of the actions appearing in the plan body, that reasoning about space is implemented, suggesting that the ideas presented there could be exploited by us. The MAS has been implemented in Jadex [56] and experiments have been run on an autonomous, collision-free motion scenario in dynamic environments. The logistics task of the autonomous resources (dumper) is to transport spoil automatically from one point of the construction side to another point as an iterative action. Unfortunately, the code is not available to the community: albeit very inspiring, we have not enough information to integrate this approach in VEsNA.

Males and Zarnic [32] represent spatio-temporal knowledge using a directional orientation approach and two out of eight relations Left and Front given by Freksa [57], plus the Meet relation introduced in Ribaric and Hrkac [58]. They integrate this spatio-temporal knowledge and the BDI modalities in a quantified modal logic. A simulation of an example in the traffic domain is implemented in NetLogo extended with two libraries that implement BDI concepts. How the gap between the modal logic and the NetLogo implementation is filled, is not explained, and the NetLogo model is not available.

Vahidnia et al. [36] implemented a logic engine named SISMORA, driven by 91 rules and axioms for the spatial and motion management. To give the flavor of how these rules look like, we show one of them taken from the deduction system for motion relations:

| MR9 | $\forall x \forall y \forall z \forall p (MFR(x,y,p) \land MT(y,z,p)) \rightarrow MT(x,z,p)$ |

In these relations, taken from the qualitative motion path calculus (QMPC) proposed by the authors and based on the qualitative trajectory calculus (QTC [59]), M stands for moving, F for following on, R for reachable. An object may also relatively move away from (MA) or towards (MT) the other. For example, MR9 says that if x is following y on a motion path p and y is moving towards z on p, then x is also moving towards z on p. Axioms and rules for means-end reasoning have been proposed, and the BDI interpreter has been re-implemented by the authors. These BDI agents are seamlessly integrated in an architecture where spatial information is managed via a GIS, and experiments were run on a taxi fleet scenario. Although very articulated, the lack of code and the extremely ad-hoc architecture suggest that we could not easily reuse it for our scope.

Finally, in a sequence of papers published between 2016 and 2018, Othmane et al. [38, 39, 40] describe a recommender system based on the BDI model, where a spatio-temporal representation of beliefs and desires is proposed. A spatio-temporal belief is an event defined as a spatial relation holding in a temporal interval. The eight binary topological predicates of the RCC calculus are generalized for modeling fuzzy topological relations, in order to represent

an imprecise spatio-temporal belief or desire such as "An accident occurred around 8 PM between road A and road B" or"I want to be at work before 9 am". An empirical evaluation of the proposed model in the traffic domain has been addressed using NetLogo extended with BDI libraires. Being implemented in NetLogo, although the code is available no Jason-like representation of space and plans is provided, making this approach not adaptable to our needs.

Since only 7 papers deal with BDI agents and spatial reasoning, and they turned out to propose approaches that cannot to be ported to VEsNA, we expanded our search to the integration of spatial reasoning into Prolog and, more in general, Logic Programming. Below, we briefly summarize the main findings of our analysis of all the 22 papers.

Table 1 classifies the 22 papers according to the high level formalisms and implementation languages involved.

The 11 papers tagged with LP in the **Model. lang.** column use Logic Programming-style facts and rules for representing spatial concepts and for reasoning on top of them. 3 of them, directly use SWI-Prolog as implementation language. 1 uses AgentSpeak. In 6 cases, there was no explicit reference to the Prolog (or "Prolog-like") language used in the implementation, in 1 case the implementation language was hard to guess from the paper, even in general terms. Finally, in one case LogC-QSD, a logical approach to compose qualitative shape descriptors defined in the paper, was used. It is worth to notice that LogC-QSD is built on top of SWI-Prolog.

The 5 papers tagged with CLP adopt Constraint Logic Programming (one adopts both LP and CLP). 4 of them use CLP(QS) as declarative reasoning framework. CLP(QS) was proposed and then used by Bhatt et al. starting from 2011, and is capable of representing and reasoning about high-level, qualitative spatial knowledge about the world. A CLP(QS) prototype has been implemented through a loose integration in C++ between SWI-Prolog and REDUCE, a tool dating back 2004 that does not seem to be available online.

Moving to the papers using less frequently used approaches, 2 papers use ASP, and only one uses ASP Modulo Theory to represent spatial knowledge and implement the reasoning upon it.

The work by Forti [47] is different from the others: it is a masted thesis, and it aims at integrating the GeoJSON format[2] into the 2P-Kt Prolog interpreter[3]. The result is named Geo2p, and it extends 2P-Kt with predicates like intersects, within, nearby.

As far as Table 2 is concerned, the most significant finding for our research is the relevance and wide adoption of the already mentioned RCC calculus used for spatial reasoning (column **Calculus**). Besides RCC, other calculi include Qualitative Constraint Network (QCN [33]), which is a graph where the nodes denote the spatial or temporal variables, and the labels on the directed edges between the nodes describe the possible relations between them; TC-6 and TC-10, two qualitative calculi for trajectories defined based on arbitrarily partitioned maps [43]; Qualitative Shape Descriptor (QSD [60]); the juxtaposition scheme QSD-Jux [61]; the Point-Line-Circuit-Area (PLCA, [62]).

---

[2]https://datatracker.ietf.org/doc/html/rfc7946, accessed on June 6, 2024.
[3]https://github.com/tuProlog/2p-kt, accessed on June 6, 2024.

We may note that case studies cover very diverse domains, and that only few papers have code associated with them (column **A.**).

To summarize the outcome of our SLR, no proposals for modelling and reasoning about space and geometry are available for the Jason that we use in the VEsNA framework. The approach closest to our needs [28] is described in AgentSpeak and implemented in Jadex, but the details provided in the paper are too few and the code is not available. The other works that mention BDI agents, either implement them in NetLogo – which is surprising, given the availability of frameworks that implement BDI agents in a way more respectful of the original BDI conception – or implement an ad-hoc BDI architecture. As far as the calculi are concerned, the survey suggests to go into the details of RCC. RCC is modelled by some authors in Prolog-like languages, that could be integrated into Jason.

5. Conclusions and Future Work

In this paper we have surveyed the literature on spatial and geometric reasoning in BDI agents and, more in general, in logic-based frameworks. We found no solution ready to be ported into VEsNA, but we obtained very interesting hints on the further literature to explore, and this is the close-future work we plan to face.

While searching for answers to our research questions, we discovered a very interesting literature field on semantic mapping, that we briefly explored starting from the paper by Chaves , Ruiz-Sarmiento, Petkov and Jiménez [63], where *ViMantic* is presented. *ViMantic* is a semantic mapping architecture for building and maintaining semantic maps of robot workspaces. A semantic map, in the robotic context, is a models that includes information about spatial elements (rooms, objects, etc.) augmented with the semantics – namely, what are the object for, what can be done with them and what cannot be done, which are the states they can be, etc. – required for an efficient robot operation [64, 65].

Although we are not dealing with robots, but with cognitive agents, the problems arising in the robotic context are very similar to ours, disregarding the fact that perception and action of VEsNA agents take place in the Virtual Reality, while robots are immersed in the real environment. While this makes a significant difference, as discussed by Davis and Ramulu [42], we are confident that sensory input received by VEsNA agents is comparable, at the right level of abstraction, with the input received by robots, and hence the methods developed for semantic mapping in robotics can be ported to VEsNA agents. Indeed, virtual environments are simpler than the real one, so we are in an advantageous position.

Chaves et al. identify the following requirements for implementing a semantic mapping:

1. A well defined model representation.
2. A mechanism for automatically populating the model representation with information coming from the environment.
3. Distributed execution.
4. Suitable human-robot interfaces.
5. Quality control.

6. Public availability.

The tool that they deem to be most suitable for representing the model, is an ontology. Chaves et al.'s vision is very inspiring, because ontologies are a key element of the abstract VEsNA architecture, although currently used at design time, and human-robot interfaces, that in the case of intelligent agents should be human-agent interfaces, are also supported by VEsNA's chatbot front-end.

The exploration of the literature on semantic mapping is hence our medium-term future work: we need to solve lower level representational problems before moving to semantics, but that is the direction we head towards.

Acknowledgments

References

[1] N. R. Jennings, K. P. Sycara, M. J. Wooldridge, A roadmap of agent research and development, Auton. Agents Multi Agent Syst. 1 (1998) 7–38.

[2] A. S. Rao, M. P. Georgeff, BDI agents: From theory to practice, in: ICMAS, The MIT Press, 1995, pp. 312–319.

[3] Open AI, Introducing ChatGPT, 2022. URL: https://openai.com/blog/chatgpt, accessed on June 6, 2024.

[4] H. Wang, H. Ning, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, M. Daneshmand, A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges, IEEE Internet Things J. 10 (2023) 14671–14688.

[5] A. Gatti, V. Mascardi, VEsNA, a framework for virtual environments via natural language agents and its application to factory automation, Robotics 12 (2023) 46.

[6] Google, Dialogflow web site, 2024. URL: https://cloud.google.com/dialogflow, accessed on June 6, 2024.

[7] Unity technologies, Unity web site, 2024. URL: https://unity.com/, accessed on June 6, 2024.

[8] T. Bocklisch, J. Faulkner, N. Pawlowski, A. Nichol, Rasa: Open source language understanding and dialogue management, CoRR abs/1712.05181 (2017). URL: http://arxiv.org/abs/1712.05181. arXiv:1712.05181.

[9] Rasa technologies, Rasa web site, 2024. URL: https://rasa.com/, accessed on June 6, 2024.

[10] Godot foundation, Godot web site, 2024. URL: https://godotengine.org/, accessed on June 6, 2024.

[11] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, Multi-agent oriented programming: programming multi-agent systems using JaCaMo, MIT Press, 2020.

[12] R. H. Bordini, J. F. Hübner, M. J. Wooldridge, Programming multi-agent systems in AgentSpeak using Jason, J. Wiley, 2007.

[13] Z. N. Esfahani, D. C. Engelmann, A. Ferrando, M. Margarone, V. Mascardi, Integrating ontologies and cognitive conversational agents in on2conv, in: EUMAS, volume 14282 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 66–82.

[14] A. Ferrando, A. Gatti, V. Mascardi, RV4Rasa: A formalism-agnostic runtime verification framework for verifying chatbots in Rasa, in: VORTEX@ISSTA, ACM, 2023, pp. 1–8.

[15] D. Aversa, S. Vassos, Action-based character AI in video-games with cogbots architecture: A preliminary report, CoRR abs/1307.3195 (2013).

[16] L. Antunes, D. Nunes, H. Coelho, The geometry of desire, in: AAMAS, IFAAMAS/ACM, 2014, pp. 1169–1172.

[17] N. YenChern, C. WaiShiang, S. KengWai, M. A. bin Khairuddin, N. bt Jali, E. ak Mit, Developing fire evacuation simulation through BDI-based modelling and simulation, Journal of Physics: Conference Series 2107 (2021) 012047. URL: https://dx.doi.org/10.1088/1742-6596/2107/1/012047. doi:10.1088/1742-6596/2107/1/012047.

[18] P. Caillou, B. Gaudou, A. Grignard, Q. C. Truong, P. Taillandier, A simple-to-use BDI architecture for agent-based modeling and simulation, in: ESSA, volume 528 of *Advances in Intelligent Systems and Computing*, Springer, 2015, pp. 15–28.

[19] P. Taillandier, M. Bourgais, P. Caillou, C. Adam, B. Gaudou, A BDI agent architecture for the GAMA modeling and simulation platform, in: MABS, volume 10399 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 3–23.

[20] E. Amouroux, T. Chu, A. Boucher, A. Drogoul, GAMA: an environment for implementing and running spatially explicit multi-agent simulations, in: PRIMA, volume 5044 of *Lecture Notes in Computer Science*, Springer, 2007, pp. 359–371.

[21] V. Marinkovic, Prolog in automated reasoning in geometry, in: Prolog: The Next 50 Years, volume 13900 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 334–345.

[22] V. Marinkovic, Argotrics - automated triangle construction solver, J. Exp. Theor. Artif. Intell. 29 (2017) 247–271.

[23] W. R. Franklin, M. Nichols, S. Samaddar, P. Wu, Geometry in Prolog, in: Advanced Computer Graphics: Proceedings of Computer Graphics Tokyo'86, Springer, 1986, pp. 71–78.

[24] W. Franklin, M. Nichols, S. Samaddar, P. Wu, Experiences with using Prolog for geometry, Graphics Inter (1986).

[25] H. Coelho, L. M. Pereira, Automated reasoning in geometry theorem proving with prolog, J. Autom. Reason. 2 (1986) 329–390.

[26] J. A. Replogle, D. J. Russomanno, A. L. D. Jongh, F. J. Claydon, Extracting isovolumes from three-dimensional torso geometry using PROLOG, IEEE Trans. Inf. Technol. Biomed. 2 (1998) 10–19.

[27] I. Sakellariou, P. Kefalas, I. Stamatopoulou, Enhancing netlogo to simulate BDI communicating agents, in: SETN, volume 5138 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 263–275.

[28] M. Schuele, P. Karaenke, Qualitative spatial reasoning with topological information in bdi agents, in: Proceedings of the 2nd Workshop on Artificial Intelligence and Logistics (AILog), Lisbon, Portugal, 2010, pp. 7–12.

[29] M. Bhatt, J. H. Lee, C. Schultz, CLP(QS): A declarative spatial reasoning framework, in: COSIT, volume 6899 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 210–230.

[30] C. Schultz, M. Bhatt, Towards a declarative spatial reasoning system, in: ECAI, volume 242 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2012, pp. 925–926.

[31] C. Schultz, M. Bhatt, Spatial symmetry driven pruning strategies for efficient declarative spatial reasoning, in: COSIT, volume 9368 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 331–353.

[32] L. Males, B. Zarnic, A spatiotemporal model of events within a BDI, in: EUROCON, IEEE, 2011, pp. 1–4.

[33] J. J. Li, Qualitative spatial and temporal reasoning with answer set programming, in: ICTAI, IEEE Computer Society, 2012, pp. 603–609.

[34] P. A. Walega, M. Bhatt, C. Schultz, ASPMT(QS): non-monotonic spatial reasoning with answer set programming modulo theories, in: LPNMR, volume 9345 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 488–501.

[35] P. A. Walega, C. Schultz, M. Bhatt, Non-monotonic spatial reasoning with answer set programming modulo theories, Theory Pract. Log. Program. 17 (2017) 205–225.

[36] M. H. Vahidnia, A. A. Alesheikh, S. K. Alavipanah, A multi-agent architecture for geosimulation of moving agents, J. Geogr. Syst. 17 (2015) 353–390.

[37] C. Schultz, M. Bhatt, Constructive geometric constraint solving as a general framework for kr-based declarative spatial reasoning, in: AAAI Workshop: Declarative Learning Based Programming, volume WS-16-07 of *AAAI Technical Report*, AAAI Press, 2016.

[38] A. B. Othmane, A. Tettamanzi, S. Villata, N. L. Thanh, A multi-context BDI recommender system: From theory to simulation, in: WI, IEEE Computer Society, 2016, pp. 602–605.

[39] A. B. Othmane, A. Tettamanzi, S. Villata, N. L. Thanh, Towards a spatio-temporal agent-based recommender system, in: AAMAS, ACM, 2017, pp. 1664–1666.

[40] A. B. Othmane, A. Tettamanzi, S. Villata, N. L. Thanh, CARS - A spatio-temporal BDI recommender system: Time, space and uncertainty, in: ICAART (1), SciTePress, 2018, pp. 48–57.

[41] E. Loza-Pacheco, M. Torrez-Ruiz, G. Guzman-Lugo, D. Loza-Pacheco, R. Cadena-Martínez, Prolog used to represent and reason qualitatively over a space domain, International Journal of Artificial Intelligence & Applications (2017). doi:10.5121/IJAIA.2017.8401.

[42] D. N. Davis, S. K. Ramulu, Reasoning with BDI robots: From simulation to physical environment - implementations and limitations, Paladyn J. Behav. Robotics 8 (2017) 39–57.

[43] G. Baryannis, I. Tachmazidis, S. Batsakis, G. Antoniou, M. Alviano, T. Sellis, P. Tsai, A trajectory calculus for qualitative spatial reasoning using answer set programming, Theory Pract. Log. Program. 18 (2018) 355–371.

[44] A. Pich, Z. Falomir, Logical composition of qualitative shapes applied to solve spatial reasoning tests, Cogn. Syst. Res. 52 (2018) 82–102.

[45] Y. Belhadad, A. Refoufi, P. Roose, Spatial reasoning about multimedia document for a profile based adaptation - combining distances, directions and topologies, Multim. Tools Appl. 77 (2018) 30437–30474.

[46] T. Grubenmann, J. Lehmann, Geolog: Scalable logic programming on spatial data, in: ICLP Technical Communications, volume 345 of *EPTCS*, 2021, pp. 191–204.

[47] M. Forti, Logic reasoning in BDI agents: Current trends and spatial integrations, 2021. Available at https://amslaurea.unibo.it/23426/1/forti_maicol_thesis.pdf.

[48] R. Buscaroli, F. Chesani, G. Giuliani, D. Loreti, P. Mello, A prolog application for reasoning

on maths puzzles with diagrams, J. Exp. Theor. Artif. Intell. 35 (2023) 1079–1099.

[49] ASPMT(QS) web site, 2024. URL: https://hub.docker.com/r/spatialreasoning/aspmtqs/, accessed on June 6, 2024.

[50] Scars web site, 2024. URL: http://modelingcommons.org/browse/one_model/4832#model_tabs_browse_info, accessed on June 6, 2024.

[51] A trajectory calculus web site, 2024. URL: https://github.com/gmparg/ICLP2018, accessed on June 6, 2024.

[52] Geolog ArcMap addin, 2024. URL: https://github.com/tobiasgrubenmann/geolog_addin, accessed on June 6, 2024.

[53] Spatial-reasoning web site, 2024. URL: https://github.com/ai-unibo/spatial-reasoning, accessed on June 6, 2024.

[54] U. Wilensky, NetLogo, Technical Report, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999. URL: http://ccl.northwestern.edu/netlogo/, accessed on June 6, 2024.

[55] D. A. Randell, Z. Cui, A. G. Cohn, A spatial logic based on regions and connection, in: KR, Morgan Kaufmann, 1992, pp. 165–176.

[56] A. Pokahr, L. Braubach, W. Lamersdorf, Jadex: A BDI reasoning engine, in: Multi-Agent Programming, volume 15 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, Springer, 2005, pp. 149–174.

[57] C. Freksa, Using orientation information for qualitative spatial reasoning, in: Spatio-Temporal Reasoning, volume 639 of *Lecture Notes in Computer Science*, Springer, 1992, pp. 162–178.

[58] S. Ribaric, T. Hrkac, A knowledge representation and reasoning based on Petri nets with spatio-temporal tokens, in: EUROCON 2007-The International Conference on "Computer as a Tool", IEEE, 2007, pp. 793–800.

[59] P. Bogaert, N. Van de Weghe, P. De Maeyer, Description, definition and proof of a qualitative state change of moving objects along a road network, Proc. of the Münster GI-Days, Münster, Germany (2004) 239–248.

[60] Z. Falomir, L. M. Cabedo, L. G. Abril, F. V. Morente, Measures of similarity between qualitative descriptions of shape, colour and size applied to mosaic assembling, J. Vis. Commun. Image Represent. 24 (2013) 388–396.

[61] L. Museros, Z. Falomir, L. Gonzalez-Abril, F. Velasco, A qualitative shape description scheme for generating new manufactured shapes, in: 25Th International Workshop on Qualitative Reasoning (QR) Co-Located at the 22Nd Joint International Conference on Artificial Intelligence (IJCAI), Barcelona, Spain, 2011.

[62] K. Takahashi, M. Goto, H. Miwa, Construction of a planar PLCA expression: A qualitative treatment of spatial data, in: ICAART (Revised Selected Papers), volume 9494 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 298–315.

[63] D. F. Chaves, J. Ruiz-Sarmiento, N. Petkov, J. G. Jiménez, *ViMantic*, a distributed robotic architecture for semantic mapping in indoor environments, Knowl. Based Syst. 232 (2021).

[64] C. Galindo, J. Fernández-Madrigal, J. González, A. Saffiotti, Robot task planning using semantic maps, Robotics Auton. Syst. 56 (2008) 955–966.

[65] A. Nüchter, J. Hertzberg, Towards semantic maps for mobile robots, Robotics Auton. Syst. 56 (2008) 915–926.