2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

Using Conceptual Spaces for Belief Update in Multi-Agent Systems

João M. L. Brezolin
Pontifical Catholic University of Rio Grande do Sul (PUCRS)
Posgraduate Programme in Computer Science,
School of Informatics (FACIN)
Porto Alegre, RS – Brazil

Sandro Rama Fiorini
Pontifical Catholic University of Rio Grande do Sul (PUCRS)
Postgraduate Programme in Computer Science,
School of Informatics (FACIN)
Porto Alegre, RS – Brazil

Marcia de Borba Campos
Pontifical Catholic University of Rio Grande do Sul (PUCRS)
Postgraduate Programme in Computer Science,
School of Informatics (FACIN)
Porto Alegre. RS – Brazil

Rafael H. Bordini
Pontifical Catholic University of Rio Grande do Sul (PUCRS)
Postgraduate Programme in Computer Science,
School of Informatics (FACIN)
Porto Alegre, RS – Brazil

***Abstract*—The BDI (Beliefs, Desires, Intentions) architecture is commonly used for the development of systems of agents situated in complex and dynamic environments. The BDI architecture represents a consolidated model that counts upon substantial theoretical and practical contributions. However, despite the strengths of symbolic models, there are some aspects of the cognitive phenomena that give rise to the need of a conceptual model that has to establish itself between the perceptual and symbolic levels. Taking into account that the use of conceptual representations can improve the recognition of objects as well as the agent communication process, the work presented here comprises the design and implementation of an integration of a conceptual-space level into the BDI agent architecture. Such integration is developed on top of the resources of the Jason platform and the CSML API. The evaluation of the implications of such a conceptual inference model for BDI agents includes the development of an application directed to the aid of users who are blind or visually impaired, as briefly discussed in this paper.**

***Keywords*—BDI architecture; Conceptual spaces; CSML API; Jason;**

I. Introduction

The beliefs of an agent consist of the way it understands the environment where it is inserted and defines parameters for its actions in order to get its goals achieved. Typically, BDI agents represent their beliefs through predicates that express properties of objects identified in the environment. However, the complexity associated with cognitive phenomena gives rise to the need for a mechanism to organize their perceptions. The process of object recognition is an example. Regarding the development of a system to support the mobility of blind or visually impaired users, it can be said that the process of object recognition is one of the essential building blocks on which this system can be implemented. This process builds the basis to establish a communication process aiming to support the mobility of those who are blind or visually impaired. The development of a level of knowledge representation based on the theory of Conceptual Spaces is one way of allowing this organizational structure can be established in the agent's architecture.

Proposed by Peter Gärdenfors [8], conceptual spaces seek to establish the interface between symbolic and perceptual levels through the implementation of a conceptual level that is founded on the idea of similarity. Conceptual spaces can be thought of as geometric spaces, in which points represent objects and regions represent concepts. The dimensions of a conceptual space denote the qualities in which these entities can be compared. The distance metric encodes the similarity between object/concepts. Such scheme provides an explanatory basis for different cognitive phenomena, such as classification, contextual effects, and object recognition [8].

Conceptual spaces also serve as a framework for representing similarity in computational tasks. It is particularly interesting for object recognition and classification. If one considers the dimensions of a given conceptual space as output descriptors of image processing algorithms, the classification of a perceived object is a matter of projecting it as a point in a conceptual space, then verifying which regions (i.e., concepts) contain the point. Such regions define the classification of the perceived object. However, in real-world applications, the perception might be incomplete, in the sense that it might fail to capture all features of the perceived object, producing a partial description of such an object. In conceptual spaces, the representation of objects is associated with the idea of partial vectors that index observed properties, and other properties that may be discovered in the future. Thus, object recognition is an association of properties that are progressively discovered until a satisfactory degree of similarity to a prototypical member of a known concept is reached.

Therefore, in order to improve the agent's abilities on the aspects of recognition and communication, we extend and apply the Jason framework [10] to allow the design of a system that incorporates the conceptual inference process on the agent's reasoning cycle. Besides the well-known robustness of this software regarding the development of multi-agent systems, the framework also allows the customization of architectural features of the agent's reasoning cycle. Such feature enables the integration of a level of knowledge representation

978-1-4673-9618-9/15 $31.00 © 2015 IEEE
DOI 10.1109/WI-IAT.2015.214

based on conceptual spaces and established with the aid of the CSML API [3].

This paper presents the first results towards the integration of these resources and its contribution to the implementation of such a conceptual level. First, we review the main aspects of the BDI architecture and the theory of conceptual spaces and present the computational platforms used our implementation. Next, we detail the customizations implemented on the existing platforms to enable our proposed conceptual inference integration and present a practical example. In the end, we draw conclusions from this study.

II. Architecture and Jason Framework

Rational agents are defined as active entities, able to analyze and act on the environment in which they are inserted. The actions performed by the agent are the result of the reasoning process guided by the goals to which they are committed. To that purpose, agents have a library of plans that can become applicable as certain environmental conditions are detected. The BDI (Beliefs, Desires, Intentions) architecture is a commonly used approach for the development of systems of agents situated in complex and dynamic environments [7]. The BDI architecture established the basis for the development of the AgentSpeak(L) Language[1].

The Jason Framework provides a platform for the development of multi-agent systems [10]. The platform was developed based on an extended version of the AgentSpeak language that incorporates support for inter-agent communication through performatives based on the theory of speech acts. Following the ideas from the AgentSpeak language, also in the Jason variant, an agent is defined through the specification of the initial state of its belief base and its plan library. The agent's beliefs represent the information the agent currently has about the environment where it is situated, desires represent the states the agent aims to achieve, and the adoption of plans in order to do so are referred to as intentions. In Jason, an AgentSpeak plan is defined through the following syntax:

$$+!g \;:\; \varphi \leftarrow p_1; \ldots; p_n.$$

where $+!g$ indicates a triggering event for the *addition* (+) of a goal $!g$; φ indicates the context that makes the plan applicable, and each p_i may be: an addition or deletion operation for beliefs, an action, a sub-goal, or a test goal. The plan context must be a logical consequence of the agent's beliefs for the plan to become applicable for handling an actual event if it matches the plan's triggering event.

In addition to supporting the basic syntax of the language AgentSpeak, Jason has support for additional functionalities such as the customization of architectural features of the agent's reasoning cycle. Such feature enables the integration of a level of knowledge representation based on conceptual spaces. Using this feature to integrate conceptual spaces in BDI agents is, as will be shown in this paper, one of the main contributions of our work.

III. Conceptual Spaces and CSML Language

Founded on principles of cognitive semantics, the paradigm of conceptual spaces proposes a model to represent concepts and similarity relations through a multidimensional geometric space. Conceptual spaces can contribute to establishing the mapping between beliefs in the agent's symbolic system and information received by its cognitive apparatus. Conceptual space is structured by quality dimensions that correspond to the mechanism used to accommodate perceptions captured by the cognitive apparatus of the agent. Quality dimensions can be separable or integral. Separable dimensions refer to characteristics that can be assigned to objects by just one quality dimension, without the aid of any other. Weight and time are examples of domains composed by a separable quality dimension. Integral dimensions refer to characteristics assigned to objects that are composed of interdependent quality dimensions values. Integral dimensions form quality domains. The color space is an example of quality domain composed by a set of integral dimensions (i.e. tonality, brightness, and chromaticity). The allocation of points in space tends to approximate elements which have related characteristics, causing the decomposition of space in regions that are characterized by convexity and connectivity. The decomposition of space in regions is determined by the existence of a cell that intersects a set of half-plans and establishes the point containing the most prototypical element of the region.

Decomposition of space in regions provides the basis for the notions of property and concept. Propriety is a term used to assign a convex region of a domain in a conceptual space (e.g. red color region). Concepts represent a set of regions in a number of domains. For example, the concept of apple is composed by regions in domains like color, shape, texture, among others. An object is represented by a vector that indexes points in regions of the Conceptual space. This mechanism enables to infer on degrees of similarity between the objects represented in space. That is, two objects may have a high degree of similarity if the value of distance computed from the points that represent them is small in relation to a domain-specific threshold and is considered similar. Otherwise, if this threshold is exceeded, these objects may not be considered similar. It is important to remark that objects usually are represented by partial vectors that index properties that could be observed on it, and the degree of similarity must be computed from the information stored in this vector. If a satisfactory degree could not be established, new interactions may be necessary to uncover new properties on the perceived object.

Adams and Raubal [2] proposed an algebraic model for conceptual spaces where convex regions are defined as convex polytopes. This algebra laid the foundation for the development of the CSML language, which supports the hierarchical representation of the elements composing a conceptual space as presented in [3]. Along with the language specification, the authors developed an API to work with CSML files. The API provides resources to create, compare, manipulate and validate CSML content with the aid of a reasoner.

IV. Integration of a Conceptual Space Model into the Reasoning Cycle of BDI Agents

Orientation and mobility (O&M) skills help people who are blind or visually impaired to know where they are, where they want to go (orientation), and how to get there safely and independently by walking or using some means of transportation (mobility). Establishing and maintaining orientation involves a cycle of perception and action in which the actions are guided by the expectations of the subject in relation to the perceptual information that they hope to find along their path. Such information can be retrieved from its cognitive map and from their experience in a similar environment. The identification of landmarks and obstacles are examples of information that is extracted from the environment by the individual in order to establish their spatial orientation [9]. This scenario indicates the need for a system based on an agent capable of recognising the objects and establishing an appropriate communication process with its user. Our expectation is that the use of conceptual representation may improve both processes.

In that direction, our goal is to integrate conceptual spaces in a BDI agent architecture in order to improve object recognition for such agents. In our proposal, conceptual spaces sit between the symbolic belief base and the perception level of the agent. The conceptual inference process involves determining the appropriate conceptual representation to identify objects perceived in the environment. This process can be initiated at the subconceptual level where the perceptions received by the agent are mapped to domains of the conceptual space. As a result, a vector that indexes points of regions of the conceptual space is generated, establishing a conceptual representation for an observed object, which can be associated with a symbolic belief that can be manipulated by the agent. As described in Figure 1, individual projections in the perceptual level $(P_1, \ldots, P_n)$ are projected onto quality domains at the conceptual level $(D(P_1), \ldots, D(P_n))$. The set of these projections forms a concept description that comprises all the properties $(Q_1, \ldots, Q_n)$ that were identified for that object. The set of all these properties establishes the foundations for creating a belief at the symbolic level that is added to the agent's belief base.

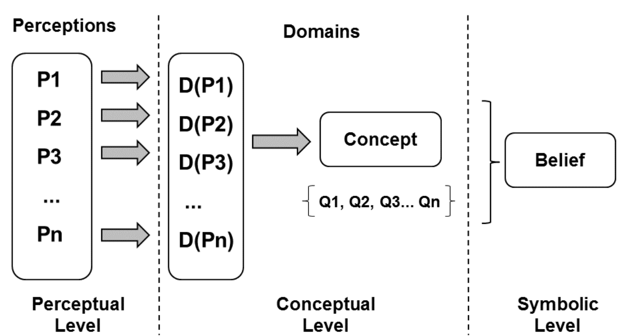

Fig. 1: Knowledge Representation Levels

On every reasoning cycle, Jason's belief-update function receives information about all that is observable in the environment, through a list of *percepts*. If a new belief literal is found in the list of percepts which is not in the belief base, this new belief is added to the latter. If a literal of the belief base (originating from perception) is no longer found in the list of percepts, it is deleted from the agent's belief base. To allow the conceptual inference, it is necessary to modify this process by adding new features to it, since it is now necessary to evaluate if new perceptions received by the agent's sensorial apparatus can compose a conceptual description of an observed object. Therefore, it is necessary to project the data captured by the sensors of the agent on domains of the conceptual space. This will allow the agent to generate a belief that represents a propriety of the perceived object. For example, when the agent receives information about the color of the object it is necessary to project it as a point in the agent's colour space. A new belief can be generated from such data allowing the agent to associate the object information with the projected point in the colour domain to which the set of perceived data refers.

Then, before this new belief is added to the agent's belief base, it is necessary to evaluate if it can, along with previous beliefs about the same object, compose a complete conceptual description. The identification of the object has, therefore, to be extracted from the new belief. Such information will be used to recover all previous properties stored in the belief base that refer to that same object. This can be done with the aid of Jason's method `getPercepts`. With the aid of the CSML API, a temporary instance of a concept can be built based on information extracted from the new (perceptual) belief and by the set of previous beliefs. This temporary instance must be compared to various concepts in the CSML specification, so as to check to which ones it could belong. With the aid of the CSML API's reasoner, a set of candidate concepts is generated by comparing the shared domains between the temporary instance and the concepts originated from CSML specification. To refine this set, the distances between the points, which represent the perceived object and the prototype elements of each one of the candidate concepts, are computed. If a degree of confidence can be reached by comparing the smallest distance computed to a previously defined domain-specific threshold, a concept description is elected. A new belief is generated and added to the agent's belief base. It is important to associate information about the conceptual space model and the instance that originated this object/concept belief as well as the information that this new belief is originated from perception. This will help the agent in recovering the object properties in future interaction processes if necessary. All those information are associated to this new belief through annotations. Otherwise, if the threshold is not met then the new belief is added as a new propriety of the perceived object (which may later help to categorise it as a member of a particular concept). This algorithm is used every time the agent receives a new perception about an object propriety. Thus, on each new reasoning cycle the agent tries to match its observation with its internal conceptual representation. The algorithm used to customise Jason's belief acquisition function is detailed in [4]. An example of the use of this algorithm is presented below.

V. A Practical Example

The previous section shows how to integrate the concept formation into the agent reasoning cycle through the adaptation of belief acquisition processes. To exemplify the use of the customised belief update method, consider the following scenario implemented through Jason's environment where an agent has the goal of recognising objects in a hotel room (e.g., to help a visually impaired user). By default, all objects are initially defined as unrecognised. The status of an object only changes when the agent classifies it. When an agent enters the room, it receives perceptions from an object. In the first interaction, the agent receives information about the color and the position of the object. The add function is called, but the information retrieved from the object is not sufficient to categorise it. Thus, both information are added into the agent's beliefs base as object properties. Although these properties do not allow the object recognition, they enable the agent to draw plans to get more information about the perceived object. Thus, the following plan becomes applicable when the agent finds an unrecognised that are near:

```
1: +unreconized(X) : hasproperty(X,position,Z) &
   near(Z)
2: <- move_towards(Z);
3:    myp.list_bels.
```

The agent recovers the object position from its previous belief and move towards to get more information about the object (line 2). On getting closer to the object, the agent receives information about the object shape (e.g., rounded shape). The add function is executed again. At this moment, the set of all perceptions (previous and current) enables the agent to identify the object as an apple. A new perception is added to the agent's belief base indicating that object refers to an instance of the concept apple. The output of the agent reasoning on this process is showed in Figure 2. The internal action `myp.list_bels` is used to show the changes on the agent's beliefs. The similarity measure computed decreases, as further object properties are perceived by the agent. In the first interaction process, the measure holds above the domain-threshold and new interactions are needed. In the second process, a value below the threshold is achieved and the previous belief that indicates the object properties are abstracted into a new belief.

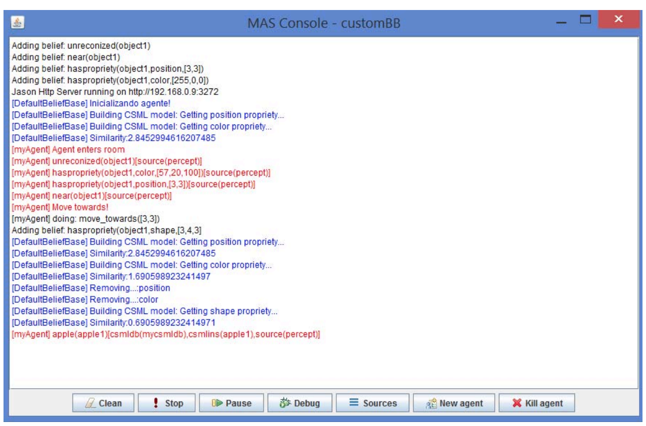

Fig. 2: Customized Add Function

It is important to remark that the customised belief acquisition function simplifies the AgentSpeak code. The whole process of determination of similarity and ability establish a conceptual generalization of an observation is left to that function (which lies at the architectural level, not the agent high-level reasoning level). In our approach, the determination of similarity tacitly induces the actions of the agent in the process of object recognition. That is, while the perceived object remains marked as unrecognised, the agent will retrieve plans from its library to try to establish the identification of the object. When a concept candidate could be elected as result of the distances calculations between the point that represents the perceived object and the prototype instances of evaluated concepts, the agent will not consider any more the perceived object as unrecognised. At that moment, the agent will able to consider it as an instance of the selected concept.

VI. Conclusion

This paper presents the first results of a comprehensive study that aims to evaluate the use of conceptual representations based on the principles of the conceptual spaces applied to Multi-Agent Systems. We seek to evaluate how these representations can contribute to improving the process of object recognition and qualify the communication process between BDI agents and visually impaired people. For the object recognition process, conceptual representations can contribute to establishing a strong foundation for the agent's beliefs, since the determination of similarity is decisive in the process of object recognition. The conceptual spaces infrastructure provides a natural way for determining the similarity with distance functions. We explore the characteristics of the agent's reasoning cycle to implement a dynamic process that is guided by the agent's actions on the environment. In our approach, we used the Jason framework and the CSML API to build the necessary infrastructure to establish a process of conceptual inference for BDI agents. We proposed and demonstrated the use of an algorithm that establishes the similarity of concepts representations based on the use of a function that computes the distance between a concept representation of an observation and prototypical instances of concepts candidates.

References

[1] A. S. Rao: AgentSpeak(L): BDI Agents Speak out in a Logical Computable Language. Proceedings of the 7th European Workshop on Modelling Autonomous Agents in a Multi-agent World : Agents Breaking Away, (1996).

[2] Benjamin Adams and Martin Raubal, *A Metric Conceptual Space Algebra*. In: Spatial Information Theory, 9th International Conference (COSIT), 2009.

[3] Benjamin Adams and Martin Raubal, *Conceptual Space Markup Language (CSML): Towards the Cognitive Semantic Web*. In: International Conference on Semantic Computing (ICSC), 2009.

[4] João Mario Lopes Brezolin and Sandro Rama Fiorini and Marcia de Borba Campos and Rafael H. Bordini, *Using Conceptual Spaces for Object Recognition in Multi-Agent Systems*. In: Proceedings of the 18th Principles and Practice of Multi-Agent Systems Conference. Bertinoro, Italy, 2015.

[5] Luc Steels and Jean-Christophe Baillie, *Shared grounding of event descriptions by autonomous robots In: Robotics and Autonomous Systems*, 2003.

[6] Michael E. Bratman, *Intention, Plans, and Practical Reason*, Cambridge, USA: Harvard University Press, 1999.

[7] Michael.J.. Wooldridge, *Introduction to Multiagent Systems*, New York, USA: Wiley and Sons, 2002.

[8] Peter Gärdenfors, *Conceptual Spaces: The Geometry of Thought*, Cambridge, USA: MIT Press, 2000.

[9] R. G. Long and N. A. Giudice, *Establishing and maintaining orientation for orientation and mobility*. In: Foundations of orientationand mobility. New York: American Foundation for the Blind, 2010.

[10] Rafael Heitor Bordini and Jomi Fred Hübner and Michael Wooldridge, *Programming Multi-Agent Systems in AgentSpeak Using Jason*, New York, USA: Wiley and Sons, 2007.

[11] Renata Wassermann, *Revising Concepts In: Fifth Workshop on Logic, Language, Information and Comunication (WoLLIC)*, 1998.

[12] S. Edelman and R. Shahbazi. *Renewing the respect for similarity*. Front. Comput. Neurosci. 6:45, 2012.