2017 Brazilian Conference on Intelligent Systems

Use of conceptual representations based on conceptual spaces theory applied to BDI agents

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

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

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

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

***Abstract*—The BDI architecture is one of the best known approaches for the development of agents situated in complex dynamic environments. Founded on the symbolic approach to AI, it is a consolidated model that builds upon substantial theoretical and practical contributions. However, there are aspects of the cognitive phenomena that require the representation of concepts to classify perceptual information in ways that have not been sufficiently explored in the BDI architecture. Taking into consideration that conceptual representations can, for example, improve object recognition processes, this paper addresses that problem, drawing from the theory of Conceptual Spaces. The paper also illustrates the viability of the proposed approach through the development of an application that aims to support visually impaired people. This application was developed using the Jason agent platform and the CSML language.**

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

I. Introduction

The BDI (Beliefs, Desires, Intentions) architecture is a consolidated model used largely to define systems of rational agents situated in complex and dynamic environments. The BDI architecture emphasises the role of beliefs in the agent's reasoning process. The agent's beliefs regarding objects observed in the environment are the result of processing a set of perceptions that relate to each other in order to establish a conceptual representation. This process requires a structure to organise the agent's perceptions so as to provide an adequate symbolic representation of real world objects. Furthermore, this structure should provide resources to support the classification of the perceived object. In this sense, in order to qualify the deductive process performed by the agent, we propose the integration of a conceptual level based on the theory of conceptual spaces into the agent's reasoning cycle, in order to assist in the process of identifying and classifying objects perceived within indoor environments. Proposed by Peter Gardenfors [16], conceptual spaces seek to establish the interface between the symbolic and perceptual levels through the implementation of a conceptual level that is founded on the idea of similarity.

In order to validate this proposal, we have developed a prototype application to assist in the mobility of visually impaired people. The prototype was developed on the Jason framework, which allowed us to implement the integration of a conceptual space to represent objects to the agents reasoning cycle. The conceptual space was designed with the aid of the CSML API that supports the definition of elements according to the algebra proposed by Adams and Raubal [4]. The prototype that we developed forms a system divided into two modules. The interface between the modules is managed by the ROS framework [14]. The first module is an application installed on an Android smartphone responsible for capturing the images of the environment. The second module comprises a Jason agent responsible for receiving the information captured by the mobile device and performing the object classification process.

This paper presents the results of an experiment on building a conceptual inference model based on conceptual spaces for BDI agents and its contribution to the development of a system to support the mobility of visually impaired users. This paper is organised as follows. First, we present our proposal for the integration of a conceptual level into the BDI agent architecture as implemented in Jason. In the next section, we briefly discuss the issues about orientation and mobility of visually impaired people that motivated the development of features included in the application. Then we present the role of the computational platforms used in our implementation. Next, we present the evaluation of the prototype by describing experiments carried out with blind users. Finally, we draw conclusions from this study.

II. A Conceptual Level for BDI Agents

In order to improve the agents' abilities on aspects of object recognition, our work proposes the integration of a level of conceptual representation into the architecture of BDI agents. In particular, we extend and use the Jason framework [18] to support the design of a prototype application that incorporates a conceptual inference process into the agents reasoning cycle. The Jason framework provides a platform for the development

978-1-5386-2407-4/17 $31.00 © 2017 IEEE

DOI 10.1109/BRACIS.2017.55

of multi-agent systems [18] based on an extended version of the AgentSpeak programming language. In Jason, an agent is defined through the specification of the initial state of its belief base and its plan library. The agent's beliefs are represented as symbolic predicates and they determine the way the agents understand the environment where they are situated.

Conceptual spaces are multidimensional geometric spaces where concepts are represented by sets of regions and where objects can be represented as point vectors. According to the conceptual spaces theory, property is a term used to designate a convex region of a domain in a conceptual space. Concepts represent a set of regions in a number of domains along with designations of salience weights and information about how these regions correlate. Conceptual spaces are structured by quality dimensions that form domains endowed with a geometric structure and a specific metric. The allocation of points in space tends to approximate elements which have related characteristics, causing the decomposition of space into regions that are characterised by convexity and connectivity. The decomposition of space into regions is determined by the existence of a cell that intersects a set of half-planes and establishes the point containing the most representative element of the region. The most prototypical element of these regions tend to stay closer to the center of the region, and the least representative elements tend to stay further away, near the intersection of that region with other regions of the conceptual space. With this metric, it is possible to infer on degrees of similarity between the objects represented in the space. That is, two objects have a high degree of similarity if the value of the distance computed from the points which represent them is small in relation to a domain-specific threshold. Otherwise, if this threshold is exceeded, these objects are not considered similar.

Adams and Raubal [4] proposed an algebraic model for conceptual spaces. According to that formalism, convex regions are defined as convex polytopes, an n-dimensional polygon representing the intersection of a set of half-spaces. This algebra laid the foundations for the development of the CSML language that allows the hierarchical representation of the elements composing the conceptual space. Along with the language specification, the authors developed an API that provides resources to create, compare, manipulate, and validate CSML content with the aid of a reasoner. Together, Jason and the CSML language provide the grounds for us to develop the integration proposed in this work.

In our proposal, conceptual spaces sit between the symbolic and the perceptual levels of the agent and are responsible for determining the appropriate conceptual representation to characterise objects perceived in the environment. The conceptual inference starts when received percepts about a given object are projected into the conceptual space (Figure 1). This projection generates a vector of points for domain regions of the conceptual spaces that indicate characteristics observed in the perceived object. In terms of conceptual spaces, this vector represents an concept instance that must be associated with a concept region. Domains of colour and shape were defined to classify objects in the conceptual space. The colour domain was defined using the RGB model; we specified regions to represent colours and we defined their prototypical elements. The shape model was defined using Hu-moments, which are defined as a vector of seven values that are invariant on scaling, rotation, and translation; Hu-moments can be used to characterise uniquely the shape of objects [8]. In both cases, similarity can be obtained by calculating the distance between the points of these vectors. Hence, these two domains allowed us to generate a ten-position vector to characterise the elements of the conceptual space that we use to categorise objects.

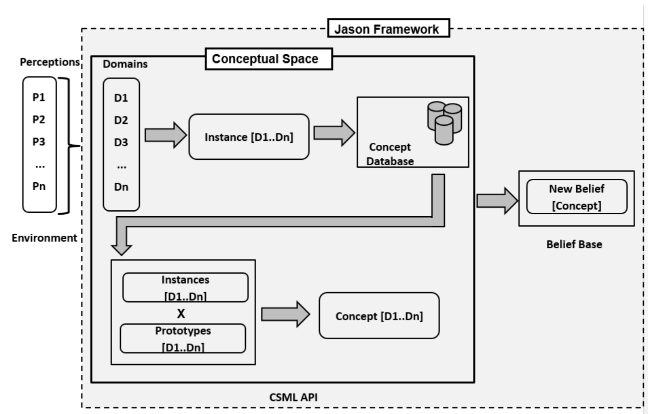

Fig. 1: Agent's architecture

The concept database was formed on top of several images of the objects that might be used as landmarks for the user. The segmentation of the objects in these images and its later feature extraction were carried with the aid of the OpenCV Library [3]. The values obtained from the extraction were used to build the concept database. A set of values were captured from each object and used to define the concept regions. For example, to build the concept of a "round table", we segment and extract the characteristics of this object from a set of images. This process generates a set of values that were used to define the shape and color regions that were used to form the concept of a "round table" in the conceptual space. To create the regions, the mathematical software q-hull [6] was used.

The object classification proceeds after the creation of a temporary instance by comparing it with prototypes of the concepts stored in the concept database. In this process, the distances between the points that represent the temporary instance and the points of the prototypes from the CSML database are computed and compared. The code used to execute this task is detailed in Algorithm 1 (Figure 2).

The first step of the algorithm evaluates if the image that contains the object is valid (line 3). As the user will be moving while the images are captured by the smartphone, it is common to get blurred images or images with no objects. Hence, invalid images are discarded before the execution of the classifier (line 5). Otherwise, the classification process is started. The object classification process assumes that the object that was captured represents an instance that must be associated with some of the concepts stored in the CSML database (line 7). To identify which concept best represents the instance, its distance from the prototype of each concept is calculated. The prototypical elements of each concept are represented by the vector of points that contains the centroid of the regions that compose it. The centroid points of each concept are extracted and the distance between its points to the points of the instance are computed (lines 10 to 12). Then, the result is stored in a temporary structure containing the concept identifier and the value of the distance calculation (line 13). At the end, the concept associated with the smallest distance value is selected (line 15). It is checked if this value is below a pre-established limit that constitutes a cut-off point (line 16). If so, the concept is chosen to represent the object; otherwise, the object will be marked as unclassified (line 17). If a concept was selected in this process, then a new belief with a reference to the determined concept must be added to the agents belief base. To do this, a message is generated at the conceptual level and sent to the symbolic level. By processing this special message, the agent will be able to update its belief base.

Algorithm 1 Object classification process

```
1:  function ObjectClassifier(Register m)
2:      Elected ← null
3:      VerifyFrameObject(getImage(m))
4:      if image is not suitable then
5:          Elected ← Not Valid
6:      else
7:          i ← createInstance(extractColor(m), extractShape(m))
8:          C ← getConcepts()
9:          for all c ∈ C do
10:             ip ← getPoints(getColorPoint(i), getShapePoint(i))
11:             cp ← getPoints(getColorCentroid(c), getShapeCentroid(c))
12:             d ← distance(ip, cp)
13:             K ← candidates(c, d)
14:         end for
15:         Elected ← ClosestConcept(K)
16:         if (getDistance(Elected) > threshold) then
17:             Elected ← Not Classified
18:         end if
19:     end if
20:     Return Elected
21: end function
```

Fig. 2: Classification algoritm

Thus, the classification algorithm creates the necessary support for Jason to establish the reasoning process at a symbolic level based on the information originated from the conceptual space.

III. Orientation and Mobility

Orientation and mobility (O&M) skills help people who are blind or visually impaired know where they are, where they want to go (orientation), and how to get there safely and independently by walking or using some form of transportation (mobility). The identification of landmarks and obstacles is an example of information that is extracted from the environment by the individual in order to establish their spatial orientation [15]. The objects perceived in the environment are identified by their peculiar characteristics and by their position in the physical space. Both types of information are used to help those who are visually impaired in the development of their cognitive map. Establishing and maintaining orientation is 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 previous experience in a similar environment. Furthermore, information previously stored in the subject's cognitive structure can establish the basis for developing cognitive maps of new places. In this sense, we have considered important to add to the message that contains the object classification, information about the position of the objects that were classified and information to help the users identify the environment where they are situated.

In this sense, after the object classification, the calculation of the position of the object in the captured image is carried out. First, the image is divided into three areas that represent the right, centre, and left positions. Then, the coordinates of the objects centre of mass are obtained. If the coordinates of the object's centre of mass are in the area that represents the center of the image, then the system infers that the object is in front of the user. If they are in the area that was designated as the left side of the image then the X coordinate is updated by adding the value of half the size of the object to it. The coordinates are checked again to check if the position of the new point remains in the area that represents the left or has been moved to the center of the image. A similar process occurs if the centre of mass is firstly found in the area that represents the right side of the image. In this case, the value of the X coordinate is updated by decreasing from it the value of half of the size of the object. After this process, the message that will be sent to the symbolic level is updated with the information about the position of the object. It is important to note that even if an object is marked as "unclassified", its positioning is calculated and the information is sent to the symbolic level of the agent. This information is considered useful for the user to identify obstacles.

After that, the system proceeds with the environment classification. The classification only occurs after the last object found in the image has been classified. Until that occurs, the result of the classification of the other objects that were captured is stored in a temporary structure. It is important to note that unclassified objects are not added to this structure. To establish the comparison, a database of relevant environments was created. In this database, each environment is represented by a tuple containing a unique identifier and a vector that contains concept identifiers. These concept identifiers represent landmarks that might be used to help user mobility. If a concept identifier is associated with an environment, it should not appear as a landmark for other environments. Each environment is also associated with a limited number of conceptual identifiers. The environment classification is the result of the comparison of the temporary structure that stores the objects classified in one image with the elements of the environment database. In this comparison, it is checked if the concept descriptions stored in the temporary structure are found in the elements of the environment database. The environment that has the largest number of matches is selected to classify the type of environment where the user is likely to be situated (e.g., a bedroom or an office).

When the objects found in an image are classified, a message containing the result of this process is sent to the symbolic level of the agent. The agents belief base is updated with the object description and its location in the scene. However, the agent only sends messages to the mobile device after the last object of the scene was classified. When this happens, a plan is fired and the messages start to be sent to the mobile device. To facilitate the understanding of the user, we chose to organise the messages in relation to the positioning of the objects in the image. First, all messages about objects located in the left area are sent to the mobile device, then the messages about the objects in the centre, and finally messages from the objects located to the right-hand side. In the end, a message containing the description of the environment is sent. The messages about objects inform their description, colour, and position. The colour of the object is informed because it is possible there are similar objects with different colours and the user might be aware of that. The final message assigns the environment's description. For example, if the system identifies only a blue door that is used as a landmark for a meeting-room environment, the message that will be reproduced by the Android device will be: "Object found: blue door. Position in front. Environment similar to meeting room." Otherwise, if the system does not get enough information about an object the following message will be reproduced for the user: "Object not identified. Position in front." The code snippet below presents the agents plan implemented in Jason (Figure 3). The first line the plans triggering event indicates that the environment was classified that is, it is *not* "not-determined" (ND). In the third line, a list of all objects positioned to the left-hand side of the image is recovered from the agents belief base. If the list is not empty, the process proceeds by iterating over the elements of that list (line 5). To each object, its description, position, and colour are also recovered from the belief base. With this information, it is possible to compose the message that will be published (lines 8–10). The same process occurs with the objects in the other areas of the image. Finally, at the end of this plan, a message containing the environment description is composed and published (lines 14–15).

```
 1: +!verify_enviroment(A,E): not(E=="ND")
 2: <- myp.getObjectFrame(A,F);
 3:    -+lastFrame(F);
 4:   +hasSimilarity(F,E);
 5:   .findall(Id1,hasObject(F,Id1)& hasPosition(Id1,Pos1)& Pos1=="left
   ",I);
 6:   if ( .length(I,Tam1) & Tam1 > 0) {
 7:       for ( .member(Oid1,I) ) {
 8:         ?hasConcept(Oid1,Con1);
 9:          myp.getDescription(Con1,Desc);
10:         .concat("Object found: ",Desc,"position: left",Msg2);
11:         pub(Msg2);
12:         .wait(500);
13:         }
14:      }
15:     ...
16:    .concat("Environment similar to ",E,Msg1);
17:     pub(Msg1);
```

Fig. 3: Jason Plan for Informing the User

IV. Technologies used to support the system

In order to keep the system easy to use and preserve the robustness of the object classification process, the system was organised into two modules. The first module is referred to as "the user module" and comprises an application installed on a smartphone. This application is responsible for capturing and transmitting images to a server on which the second module of the systems is installed. The use of a mobile device to capture images aims to facilitate user acceptance of the system since this device might be part of the blind user's daily life. To make use of the application, the user directs the device's camera to the location and waits for the application to start reproducing the messages sent by the server. The messages contain information about the objects found and help the user to figure how the environment is organised. The process of exchanging messages between the smartphone and the server is fast, it does not take more than a few seconds.

In the server, the second module of the system, called "Intelligent System Module", is installed. It is responsible for the classification of the objects identified in images sent by the smartphone. At the end of this process, this module is also responsible for sending back to the smartphone messages that reflect the result of this process. As soon as the messages are received by the mobile device, they are reproduced through its sound interface.

The interface between the modules of the system occurs through ROS (Robot Operational System) [14]. ROS is a framework that provides a layer of abstraction of hardware and services that help in the development of applications in robotics. ROS uses nodes to encapsulate processes that perform computations. These processes communicate with each other through messages that are exchanged through topics. In the "User Module", two nodes were implemented: the first one is responsible for the transmission of the images captured by the mobile device to the server. The second node is responsible for reading the messages that will be published by the "Intelligent System Module". This module is responsible for processing the images received by the mobile device. This processing starts at the moment that the image captured by the mobile device is received by the node installed on the server.

The process of object classification comprises three levels. At the perceptual level, the feature extraction process is performed with the aid of the OpenCV library which provides resources for segmenting and extracting data from the objects identified in the image. The extracted data is sent to the conceptual level, by using a ROS node that is read by a Jason-ROS [7] interface artifact. This artifact was developed on top of the CArtAgO framework [2] and was integrated into the system to help with sending and receiving messages on ROS nodes.

After receiving the message from the perceptual level, the Jason agent executes the projection of the data into the conceptual space and performs the classification of the objects.

The result of the classification process is sent to the symbolic level of the agent. At this level, the agent organises the messages as previously described (see lines 10 and 15 of Figure 3) and publishes it in a node that is accessed by the smartphone (Figure 4).

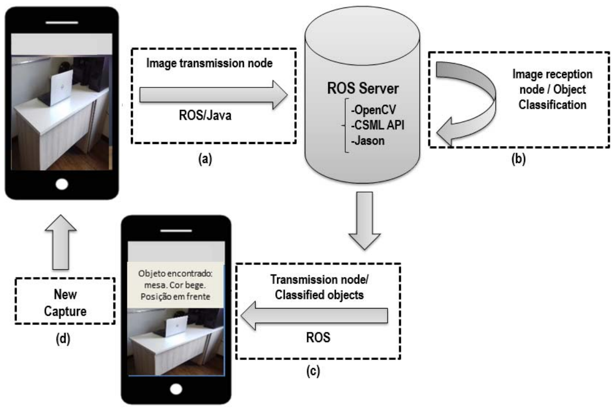

Fig. 4: System organization

The text of the message is reproduced for the user by the Android text-to-speech engine. The Android device also fires a vibration to warn the user about a presence of the object detected.

V. Evaluation of the prototype

Two experiments were conducted with blind users to evaluate the usability of the system and the accuracy of the classifier based on conceptual spaces. The experiments were performed in different places as well as executed by different users. The intelligent system module was installed on a Ubuntu server version 14.04 on a virtual machine with 4 GB of RAM. The host of this virtual machine is a Dell Inspiron 15 Notebook with an iCore 7 processor and 16 GB of RAM. The user module has been tested on a Samsung Galaxy Prime Gran Duos device with 8 GB of RAM and with the Android Operating System version Lolipop 5.1 installed.

Before conducting the experiments, an interview was carried out with the users to identify their profile and their expectations regarding the use of the application. A post-test interview was also carried out to evaluate the impressions of the users about the use of the application. The interviews were composed of semi-structured questions that allow the users to speak freely about their experience [9]. The interviews and the experiments were recorded on audio and video.

The first experiment occurred in an office where the user had the task of finding some objects in this environment. It is important to highlight that the user did not have any previous knowledge about the environment or about the objects that he would find. The user selected for this experiment is blind from birth and is between 18 and 25 years old. The user has experience using smartphones but has never used any application to assist in moving around through open or indoors environments. In the office where the experiment was conducted, the objects were distributed as follows: on the left there was a shelf; in front of the door, about 2 metres, there was a black chair in front of the user; on the right (about 1 metre), there was a circular table and on it there was a laptop computer. Beside the door there was a backpack resting on a shelf. We opted to add elements such as a laptop and a backpack to demonstrate that the classification performed by the system could assist not only in the identification of furniture but also of daily-use objects.

Initially, the user identified the chair and then the shelf when moving towards the chair. The user was able to identify the backpack object with the exclusive use of the application and the rounded table with the use of the application combined with tact. After moving around the environment, the user was asked to create a concrete representation the environment they had just explored. The user received miniature materials representing objects and furniture in the room and distributed them on a tactile map (Figure 5). All of the objects were positioned correctly.

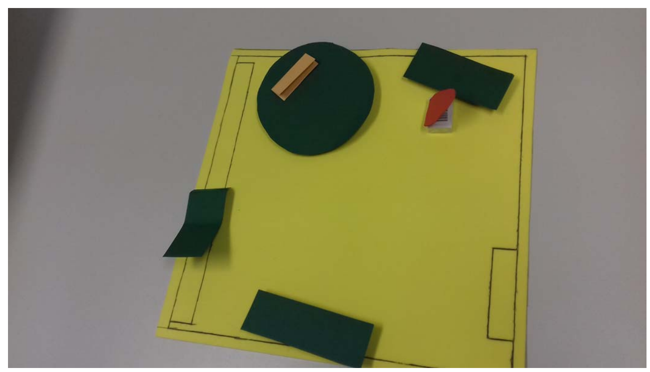

Fig. 5: Tactile map

The second experiment was carried out with a different user in a meeting room. As in the previous experiment, the user had the task of finding a limited number of landmarks in the environment. The second user has become blind during adolescence and is in the age group between 36 and 45 years old and has limited experience in using smartphones. In the meeting room, the user could identify six objects: table, laptop, window, sound box, chairs, and window shade. When the user started using the application, the system identified the chairs and the window shade. From this information acquired through the use of the application, the user was able to move over the environment. Finding the window shade helped the user to return to the hallway and move forward. At the end of the hallway, the system identified a laptop computer. The table was not identified by the application, but the user assumed that the laptop was on it. It was initially observed that the user presented a certain difficulty in understanding that the system informed the position of the objects in relation to the position of the smartphone camera and not in relation to his body positioning. Even so, the user was able to proceed with the exploration of the environment.

According to the first user the system, it was easy to use and helped him understand the objects and furniture present in the environment; the sound interface was useful to understand this relationship. For the second user, the system was also easy to use. The user, however, reported difficulties with camera positioning. He considered the use of the sound interface and the vibration feature important to understand the relationship between objects and furniture.

In this experiment, 214 classifying processes were performed and recorded while the user was exploring the environment. In these processes, the system tries to identify if the object captured matches one of the six possible landmarks. From the 214 classification processes, 134 had results that were filtered by the classification algorithm. In these cases, the distance between the point that represents the observed object and the centroids of the concepts was above the cut-off point established in the classification algorithm. Much of this result is related to the fact that the system performs the image capture while the user moves. In this case it is quite common the capture invalid images. This explains the high number of discarded objects. From the 214 classification processes, 21 misclassifications and 59 correct classifications were recorded.

The results showed that the conceptual spaces approach was flexible and allowed the classification of objects even if they were not captured accurately or were captured only partially. We sought to point out the viability of the conceptual spaces approach. This approach was adequate for the type of recognition required for the context in which the application was used, that is, the capture and classification of objects with the user in motion and the partial identification of objects.

VI. Conclusion

This paper sought to discuss the feasibility of the conceptual space approach. This approach established the foundation to create a conceptual level to represent objects as part of a symbolic agent architecture. The conceptual level was integrated to the agent's architecture and used in a system that was used to support the mobility of blind users. The approach was adequate for the type of application being proposed. Concept-based object classification makes the system more flexible and allows the system to classify objects even when they are partially captured. The conceptual spaces approach produce good results, which might be improved with the addition of new domains to the conceptual space, which would allow the creation of better descriptors for the classification of objects. Training users before tey use the application might also contribute to better results. However, in the experiments, both users pointed out that the system was useful and allowed them to recognise objects and furniture in the environment and to establish the relationship between them. To facilitate the use of the application by the user, smartphones are used to capture images of objects. The users mentioned that this decision was appropriate because it is easy for them to adapt to the use of an application on the mobile device.

The classification of objects using the projection of objects and concepts in the conceptual space was also adequate. Through the classifier, it was possible to identify most of the objects in the environments and establish a good cut-off point for the misclassifications. The system presented a relatively fast classification process by sending information to the user within a short time.

The Jason framework allowed establishing a natural interface between the symbolic and conceptual levels, which creates a potential use of this approach in different applications of multi-agent systems. An example is the use of conceptual spaces to qualify communication between agents. In general, this research sought to establish the basis for new research to be carried out in relation to the use of conceptual spaces to represent perceptual knowledge in BDI agents and for the potential of this approach to be explored in several application areas.

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] Alessandro Ricci, M. Viroli, and A. Omicini.*CArtAgO: An Infrastructure for Engineering Computational Environments*. Proceedings of the 3rd International Workshop on Environments for Multi-Agent Systems (E4MAS). 2005.

[3] Ana Huaman and others. *OpenCV Documentation*. Avaliable in: http://docs.opencv.org/2.4/doc/ tutorials/tutorials.html.

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

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

[6] C. Barber, Bradford, Dobkin, David P. and Huhdanpaa, Hannu. *The Quickhull Algorithm for Convex Hulls*. ACM Trans. Math. Softw. 1996.

[7] Felipe Meneguzzi and others. *Jason-ROS*. Avaliable in: https://github.com/lsa-pucrs/jason-ros-releases/releases.

[8] Hu Ming Kuei . *Visual pattern recognition by moment invariants*. In: IRE Transactions on Information Theory, 1962.

[9] Jennifer Preece, Yvonne Rodgers and Helen Sharp . *Design de Interação: Além da Interação Humano-Computador*. Bookman. 2005.

[10] João Mario Lopes Brezolin, Sandro Rama Fiorini, 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.

[11] L. Keyes and A. Winstanley. *Using moment invariants for classifying shapes on large-scale maps*. In: IComputers, Environment and Urban Systems, 2001.

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

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

[14] Morgan Quigley, Ken Conley, Brian P. Gerkey and others. *ROS: an open-source Robot Operating System*. Proceedings of the ICRA Workshop on Open Source Software. 2009.

[15] N. A. Giudice and G. E. Legge, *Blind navigation and the role of technology*. In: Engineering handbook of smart technology for aging, disability, and independence. New York, USA: Wiley and Sons, 2008.

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

[17] 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.

[18] 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.