Use of a Semantic Language to Reduce the Indeterminacy in Agents Communication

Horacio Paggi
Facultad de Ingeniería - LISI
Universidad ORT
Montevideo, Uruguay
Email: horacio.paggi@gmail.com

Michael Cochez
Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland
Email: michael.cochez@jyu.fi

***Abstract*— In the field of agent communications uncertainty and vagueness in the message content and in the achievable results play a primordial role when two agents (human or artificial) communicate. Even though the importance of vagueness and uncertainty has been recognized long ago, only recently mechanisms related to the communications' semantics that allow a practical approach have been designed; more specifically, the development of tools such as agent programming languages and frameworks, which is a field of intensive research. On the other hand, recent theoretical ideas, drawn from situation semantics theory and the works of Sutton on semantic information, support this work. This paper applies these ideas to the field of multi-agent systems (MAS) and sketches how one can reduce the impact of vagueness and uncertainty present in the communication between software agents by means of context information, collaboration and basic reinforcement learning using a language designed for agent communication: the Sematic Agent Programming Language (S-APL).**

***Keywords:* communications semantics, multi-agent systems programming language**

I. INTRODUCTION [1]

When two agents, human or artificial, communicate, vagueness can arise related to the meaning of the message (its semantics) which in turn implies a variable response, i.e. in the pragmatic content of the message, as is described in the speech acts theory [1, 2]. Concurrently, there are uncertainties related to the results of the actions triggered by the message in the receiver agent. These uncertainties are related to the dynamic behavior of the system and the previous performances of the sending and receiving agents. Similar to Sutton [3] we won't try to define what vagueness is, because, in the words of Austin "Vague is itself vague" [4]. As said, this vagueness is tied to the semantic interpretation of the messages and the reasoning made about their possible effects under a "common sense" assumption (a default context) or a specific context; on the other hand, the uncertainties are linked to the previous behaviors of the system (which can be considered a temporal context), and both of them to its learning capability. Note that we are not considering the problems related to a noisy communication channel: we assume the communication to be free of noise. Further, we take for granted that when a message is received, the message comes from the sender and has not been altered by a third party. Furthermore, we assume that agents have good intentions, i.e. do not intentionally provide false information. Hence, messages received contain no more information than is contained in the situation that originated the message [3]., but we will include the case in which information is lost, i.e. the "equivocations" of Sutton. The main idea here is to show the programming constructions in S-APL which allow implementing ideas taken from the philosophy of language field in the MAS communication vagueness. Many frameworks and languages that allow agent programming and communication have been developed; in the specific case of messages coded in a FIPA-ACL language we found languages like Jack, Jason, Jade, IndiGolog, etc. However, these specifications focus on the message syntax processing and do not support the above mentioned semantic interpretation. Basically, they do not allow to explicitly state the message context. In this work we show how, in the specific case of a MAS, one can reduce the negative effects that these indeterminacies can have, using a semantic agent programming language. That is, a language that can be used as content language and that also allows the specification of the behavior of the agents. This is achieved by using contexts and implementing a basic schema of knowledge sharing and reinforcement learning.

In this context, "negative effects" means the likelihood that the response of the receiver, in the form of messages returned or actions of any kind resulting from the received message, is not the one expected by the sender. Reducing these negative effect from the receiver's side will mean answering or acting in a way the sender expects, effectively

[1] *This work was partially funded by the grant S-C-BE 55/18, Préstamo BID OCUR /1296-PDT and by the PEDECIBA, Uruguay*

(that is, with positive results)[2]. In practice, however, the receiver will not know the exact expectations of the sender and will hence have to work with an approximation. For example, if in a classroom the teacher asks a student "please, close the door" and the other person closes and locks it, this would not be considered a positive answer/result to his requirement because he/she just wanted the door closed, not locked. Another example: If a solution to a "slow internet connection" problem is "reset the modem", surely this means "turn off and then turn on the modem" but not to restore it to its factory settings.

This work is structured as follows: next, we describe many basic concepts and related work, in the following section S-APL is briefly presented and we outline how problems related to indeterminacy can be handled using it; then an example (a help-desk receiving requests from users and trying to solve them) is given and, finally, the conclusions and future work are stated.

II. RELATED WORK

A. Preliminary concepts

1) Indeterminacy, uncertainty and vagueness

Indeterminacy is in the context of this paper related to the degree of knowledge one has about the immediate consequences of a message; in the words of Novák: "uncertainty and vagueness form two complementary facets of a more general phenomenon which we may call *indeterminacy*" [5]. Indeterminacy (i.e. uncertainty and vagueness) implies a degree of belief in the communicated proposition, and in turn, only one tendency to act: citing Smith, "when a term is familiar, it can be used without people asking 'what does that mean? ' … a degree of belief that proposition P [is true] implies a tendency to act as if P [is true]"[6]. We are interested in the vagueness and uncertainty just in the way they can affect the beliefs and responses of the agents.

Given a subject (an agent), vagueness arises when she/he tries to group objects with a given property; it's the opposite to exactness and cannot be avoided in the human way of regarding the world, moreover, it can be necessary in order to convey relevant information (the "incompatibility principle" of Zadeh [7]) . Vagueness can be modeled as degrees of truth [6] and is then naturally related to the fuzzy sets theory and its concept of membership function [5, 8]; it is also associated with how a phenomenon is defined (and not with its occurrence) and is typical of natural language.

Two different kind of vagueness can be found in the agents communications:

- The proper one of the vocabularies (ontologies) used in the communication. If the ontologies allow fuzzy concepts – as in the case of f-OWL [7]- , then a fuzzy concept (generally related to a linguistic variable) may have different values in its membership function μ in the sender and in the receiver: the notion of "tall person" may be given by a μ shaped as a right shoulder (0,170,180,*) – meaning that a person who is less than 170cm. high is not tall, a person over 180cm. high is definitely tall - for the sender while for the receiver it could be (0,185,195,*). It is our viewpoint that this kind of vagueness cannot be handled through classic ontology matching methods based on a terminological or structured viewpoint as proposed in e.g [9]. On the other hand, an extensional approach (comparing instances of the concepts) could take too much time. Instead, we propose the use of strategies such as considering the context of the message (for example, if Peter is 20 years old, an 'OLD MAN' may mean a 40 year old person). This kind of vagueness corresponds to the U1 uncertainty type of Sutton.

- The vagueness associated with concepts that don't match in the sender's and receiver's ontologies (the Sutton's U2 uncertainty type). Different collaborative approaches have been taken for this: in the case of S-APL, the agent can query other agents about how to proceed, in the case of CooL-AgentSpeak [10]an explicit search of the unknown concept in a set of collaborative agents' ontologies is triggered.

Unlike vagueness, uncertainty has an epistemic character [5] and can be seen as a doubt about the possible results that an event or action can have, or even the lack of knowledge about the occurrence of an event. In this work we are interested in the analysis of the uncertainty in the light of past behaviors (considering as "behaviors" the answers that the agent has given to previous messages) as in the case of posterior Bayesian probability. Randomness is a specific kind of uncertainty: the one related to time. There is no randomness after the completion of an experiment, when the results are known [5]. In our case, there will be no randomness after the reception of the answer.

Uncertainty can be modeled in several ways: probability theory, possibility theory, beliefs measures, etc. Uncertainty can be handled by having the rules used in the decision making (reasoning) process and the facts as data stored in the same container, incorporating information structures,

[2] The definition could be relaxed saying that the answer is one of the possible ones expected by the sender.

when needed, that record the probabilities of that answer is the most adequate for the case. These probabilities could be regarded as frequencies, as Sutton suggests, or as a subjective value that can be assigned (corresponding to a *frequentist* or *subjectivist* interpretation of the probabilities, respectively) of that answer is the most adequate for the case. The frequentist approach seems to be the more adequate for rational agents.

Note that we are considering a setting where the sender has no doubt on the content of the message: the message sent is certain for the sender, but can be vague, on the other hand, in a dialogue the roles of sender/receiver alternate so all we can do about the learning in the receiver agent applies to the other one (which in turn will be the receiver of the answer).

B. *Alternatives in the computational mangement of indeterminacy*

1) *Ontology matching*

The interpretation of the sent message is based on a correspondence between a certain (set of) concepts (possibly complex) in the sender's ontology and the related concepts in the receiver's one – that is, after an ontology matching. In the ontology matching field a vast amount of work has been done, and, specifically in the MAS field, several projects have been undertaken: the DOMAC system proposes a dynamic mapping based in three approaches: lexical, semantic and structural [11]; the Coo-AgentSpeak [12] and the CooL-AgentSpeak face the problem too, and finally, at a theoretical level, the Ontology Service of FIPA is designed to perform the mapping between the ontologies of the communicating agents [13]. Another approach, as described in [14] is to use a centralized messaging component that uses ontology learning and matching. This message broker is responsible for only sending information to parties involved in the communication which the parties understand. S-APL does not use any automated ontology matching procedure in order to get an alignment between the sender's and receiver's ontologies. Rather, it uses an ontology linking one, in which only the relevant concepts of the both ontologies are defined and maintained by another agent or organization in a third ontology, called upper ontology.[3] The relevant concepts are the concepts common to the both ontologies (e.g. "resource" is a relevant concept for "printer" and "activity" of "printing document" [15]. Relevant concepts are super-classes of the original ones. In this way, when communicating actions and intentions using relevant concepts they can be set coordinations (correspondences) between the ontologies which are evolving or that are not completely known at this moment [15] .

2) *Other alternatives to treat communications' indeterminacy*

The alternatives in the implementation of the indeterminacy impact reduction are not very abundant. The language Cooperative Description Logics AgentSpeak: CooL-AgentSpeak [10], based in AgentSpeak and its interpreter Jason [16] is the functionally most similar to S-APL. In this case, when an agent does not find an adequate answer to the received message, it can ask another agent for help. The latter shares with the former the needed answer(s) (if it has any), in other words, that collaborates giving a relevant plan (a series of activities answering the message). This approach allows to register the uncertainty using "mental notes", that is a record of the beliefs generated due the execution of an agent´s plan, registering the cases when a plan succeeded or failed. On a future, similar occasion these records can be investigated to choose a feasible plan. The main shortcoming of this tool is that it is not FIPA compliant so problems can arise in the compatibility with agents not developed based on AgentSpeak/Jason.

Another attempt to enable the communication at a semantic level was the development of the Jade Semantic Add-on (JSA) [17, 18]. The tool is a Jade extension, a set of classes which tries to make the coding of Jade agents simpler. It lacks the capabilities of CooLAgentSpeak (in the sense of automatic search of a subsuming concept or plan), so it provides a limited support to the complex behaviors of the agents. Also, one of its biggest drawbacks is that nowadays there is no team developing and maintaining it.
Trying to overcome some of the problems of the JSA, an architecture based in the JSA plus elements of the IndiGolog [19] (basically in the planning aspects) was developed by Lesperance and Shapiro, but its implementation is still experimental [19]

Jadex [20] is a Java-based platform that allows the development and communication of BDI agents, it features a forward chaining reasoning engine. The agents can be deployed in a middleware such as Jade; it allows sending and receiving of messages specifying the (common) ontology used and a codec to code/decode the message content, which in turn could be implemented to handle the indeterminacies in the messages. Additionally, as we'll do with S-APL, some kind of reinforcement learning could be implemented. The idea of using S-APL is to simplify the message so no codec is needed.

Py Ouyan and Fu [21] proposed a model of communication for hybrid agents in which the communication language is expressed in terms of a common ontology (described in OWL) shared by the agents.

[3] To be completely precise, S-APL is only the language. The ontology linking would be task of the platform on which S-APL is used, e.g. UBIWARE.

Finally, Gonçalves and Gluz [22] developed AgentSpeak(PL), an agent programming language based on AgentSpeak, where the knowledge of the environment can have degrees of certainty (expressed as a probability). A formal analysis of this kind of communication can be found in [23].

Note that if the concepts or intentions of the sender can't be found in the receiver's ontology, additional information is required in the receiver and not only the degree of truth they have associated.

C. Semantics a a learned probabilistic correlation

In situation theory, meaning is relational [24]. Roughly, the meaning of an expression is a relation between the speaker connections, a context and a described situation. The speaker connection is the connection (association) made between the utterance and the different objects which can refer to. More simply, Sutton states that "the meaning of an expression is a relation between a discourse situation [called the context here] and a described situation" [3]. The described situation is, in a nutshell, what is said, a situation in which the world is in some say (for example, "John is tall").

An iterated learning model (ILM) [3] is, in its simplest form, a collection of pairs (string, meaning) that are learnt through time. In this work the implementation of such ILM using S-APL is sketched. Given that not all the possible pairs (string, meaning) are presented to the learner before its execution (the bottleneck problem) we can also reduce the impact of indeterminacy in the communication using frequency recording: the receiver learns the possible connections and assigns a frequency (a probability) of appearance to them.

D. S-APL origins

S-APL was originally developed in the Smartsource and the Ubiware project [25]. One of the main motivations for its development was the need of a language that allow the explicit removal of existing information in the agents and the embedding of queries and rules as normal beliefs of the agent.

III. Implementation overview

In this section we describe how the context management and the reinforcement learning (the implementation of an ILM) can be done.

When an agent doesn't know the correct answer to a message, the Ubiware platform [15] (in which agent using S-APL reside) provides mechanisms for implementing the selection of the agent that can do it. This can, for example, be achieved with an English auction selection, where the 'price' offered corresponds to the likelihood of answering correctly to the message under consideration. Knowledge sharing can help to reduce the impact of vagueness, while the learning improves the results with respect to the uncertainty (the uncertainty of the receiving agent about the answer of the sender as being "Yes"/"No" or "Correct"/"Incorrect").

The vagueness can be handled also through the use of contexts: S-APL allows indicating the context of validity of a statement[4]. A statement that has a certain degree of vagueness (e. g. a degree of truth) in a given context might have a different one in an ampler context (contexts may form a hierarchical structure). These degrees of truth can be used to determine the consequences of the statement. The degree of truth assigned to a message is a basic pragmatic property. For example, suppose that an agent X receives a message from agent Y stating (text in `courier` font represents constructions of S-APL) "`:John :hasHeight "tall"`" then this means that the fact that John is tall is a true statement at the present time for agent Y. That is, agent Y has such believe in its global context G[5]. If the message content would have been "`{:John :hasHeight "tall"} :accordingTo :Z`", then Y makes clear that the information is provided by agent Z and may not be as true as if it were a proper observation of Y. The degree of truth or confidence assigned in X to the fact that John is tall is surely different in the former and the latter situation(for example, if Z is 1.50 meters high), which will imply different reactions in the receiver.

Additionally, it could be tested if a belief expressed in a message was obtained directly by the agent or was informed by another agent by using a query such as:
`{:John :hasHeight "tall" } according to ?x` which will return a non-empty result only if there exists another agent which informed it to the agent.

In order to quantify the credibility of Z, we could record the number of cases in which the statement is true for the agent and for some other agent:

```
{
    ?x :accordingTo :Z .
    ?x sapl:is sapl:true
} sapl:implies {ccccccc}
```

where `ccccccc` represents the statements needed to record the increase of the credibility of Z.

This recording of the number of times that a statement is true given that it comes from agent Y is a very basic form of learning. As Sutton suggests [3], mechanisms of pattern recognition could be used in order to associate not only a probability of a degree of truth of a message coming from Y but also to a pattern of messages.

[4] This is a characteristic inherited from N3logic.

Another option could be to act depending on that other agents state the same:

```
{
{:John :hasHeight "tall"} :accordingTo :Z.
{:John:hasHeight  "tall"} :accordingTo :Q}
} sapl:implies {… … …}
```

(if Z and Q state that John is tall then do…)

We are assuming here that the source of the information is not the same and that in fact there are two independent agents Z and Q saying the same. This is, Z and Q are not repeating some information they heard: there are no coalitions that drive the receiver agent to do something)
Given the reinforcement learning we are trying to implement, we would like to use the track of the tuples *(sender, receiver, message/intention, context, answer/action, number of successes)* so that we could select the most appropriate action/answer. A logical choice is using the procedure with more previous successes for a given sender and a context of the same or greater extent, or with more successes for a sender with which no prior interaction has taken place.

a) After the initial sender has sent a confirmation of a correct answer or notified about expected results of actions, increment in the successes count could be trigerred as follows:

```
{
?mes :hasSolution ?ans .
?counter :hasMessage ?mes .
?counter :hasAnswer ?ans .
?counter :hasValue ?val .
?newval sapl:expression "?val+1"
}
sapl:implies
{ ?counter :hasValue  ?newVal.
sapl:I sapl:remove {
   ?counter :hasValue ?val .
   ?mes :hasSolution ?ans
} .
}
```

The next time the receiver interacts with the same sender, for a given message, the former can use an S-APL query with the `max` function in order to select the answer with biggest number of successes.

b) A more direct (but without the elegance of the beliefs handling) would be to create a data structure (table) with (sender id, receiver id, answer id, message id, number of successes). Such table, which would be in an external storage, could be accessed using external behaviors such as the SQLReader, using "update" instead of "select" commands (see [26, 27]) to find the most frequent valid answer for the present message given the previous experiences.

Note that the checking can be done before the execution of the action that could affect adversely the sender (using an ordinary message exchange), or maybe twice: before the execution and after it, if the sender is the only one who can say that the action was completed correctly.

As can be seen, when consulting a (limited) number of agents in order to find the most probable answer, one only gets a local minimum of the impact of the indeterminacy. This is another reason why we do not speak of "to minimize the impact" but rather "to reduce the impact".

Finally, we mentioned that failed actions would be considered when evaluating the impact. This can be implemented through an English auction protocol where the price of the offer is the probability of to end the action satisfactory (in the context of the number of times the action was executed: a probability of 100% in 2 executions may be less meaningful than 90% in 2000).

IV. An example: a help desk.

Suppose we have this scenario: a help desk receives service requests from users. The requesting user connects with an agent (an interface agent) that will ask several questions in order to diagnose their problem. In the communication between the user and the help desk there is a certain degree of vagueness (e. g. "my pc runs too slow").

The help desk is formed by three components:

a) an interface agent with the user,
b) a "solver" which searches actions intended to solve the problem and
c) at least one administrator caring to keep the knowledge repositories used by the "solver" up to date by adding/deleting rules related to the domain of knowledge of the help desk.

The "solver" can be considered as formed by one or more components (human agents and software agents organized as dynamic hierarchies). Note that this is an (more or less) open structure: specific components could join the solver to solve certain problems, as when a specialist is hired temporally. The use of a tool such as S-APL which is based in Notation3 – developed for the semantic web - has the advantage of to allow to model naturally the knowledge and rules of such a component

The human agents collaborate with the software ones performing actions related with the physical world (for instance, to replace a faulty network card).

There can be vagueness in the expressions used by the user or in the questions asked, so a rules engine capable of backward chaining is needed (for example, Fuzzy Jess) In this point, vagueness is attacked with the proper tools of the rules engine.

As result of the diagnosis, a series of messages composed by the interface agent (containing some vagueness) will be sent to the solver agent describing the case and the context (user, configuration, performed tests , etc.). These messages would be coded using S-APL which in turn could be used to choose the most promising answer (remember the uncertainty about the valid answer) and generate a simple plan[6] in case of acceptance of the solution found. The use of S-APL has an additional advantage: we do not need to code a parser that interprets the content language and decides what to do.

The help desk then tries to solve these incidents in such a way that the answer (solution) is the most expected (satisfactory) for the user.

The following diagram depicts the help desk

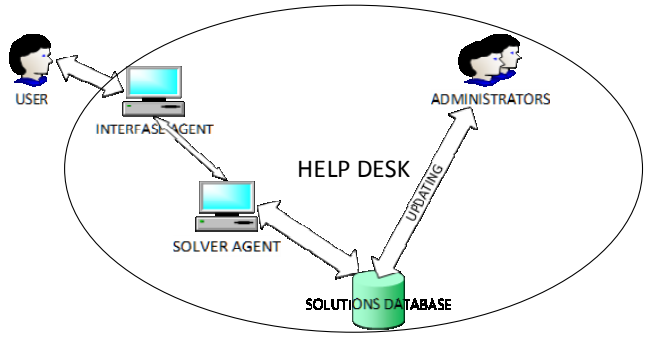

Figure 1. Help desk schema

A message sent from the interface agent to the solver could be:

```
:John :hasproblem :PCslow
```

where John is the user who is having problems with his PC speed.

The Solver agent searches the possible solutions for the incident. Its beliefs could be something like:

```
……..
:PCslow :hasSolution  :execAntivirus .
:PCslow :hasSolution  :Scandisk .
:PCslow :hasSolution  :CleanDisk .
:execAntivirus :hasEffectiveness  0.90 .
:ScanDisk   :hasEffectiveness  ?a1 .
:CleanDisk  :hasEffectiveness  ?a2 .
```

The interface agent presents the solutions found to the user who selects one and then the interface executes it, perhaps as a behavior, as suggested by the solver. The solver then asks the user if the problem was solved. If it was solved, a message

```
{:PCslow  :hasSolution  :Scandisk}
:accordingTo :John
```

is send to the solver agent so it increases the effectiveness of that solution (e.g. Scandisk), so the semantics of :PCslow when the user is John is reinforced to the meaning "Scandisk" by augmenting its success count. For this the receiver (solver) has a conditional commitment rule (among many others) of the form

```
{{?problem  :hasSolution  ?solution}
:accordingTo ?user}
=>
{?problem :hasSolution ?solution .
sapl:I sapl:remove {
   {?problem  :hasSolution  ?solution}
:accordingTo ?user
}
```

The right expression is ignored if the left is false and the right –hand side of the rule triggers other rules which increment the respective counters.

V. Conclusions and Future Work

In this work we have sketched how communication indeterminacy of agents could be handled (and its effects alleviated) using a semantic agent programming language: S-APL. A distinctive feature of it (the externalization of the beliefs in reservoirs separated from the other agents data) and the use of contexts allows the receiver to find an adequate answer, which in turn should be checked (maybe before its execution, in the case that the answer is an action, or maybe twice: before the execution and after it, if the sender is the only one who can says that the action was completed correctly) with the sender, having a loop of reinforcement learning.

Many implementation and performance related further research questions arise:

1. First of all, an analysis of the evolution of the impact of the indeterminacy over time (this is, through learning) is needed, which could be studied by building a prototype.
2. In a context of bounded time and rationality, the balance between the number of agents that can be queried and the degree of reduction in the indeterminacy could be investigated. A further open question is how this reduction could be measured.
3. Moreover, it seems that answers given in the far past are not as relevant as fresh ones. A

[6] S-APL doesn't have yet a planning mechanism for the internal actions of the agent as other tools have (e.g. Jadex), so the plan is a simple list of actions that trigger each other as long they are performed successfully.

question remains as to when certain answers should be deleted because they have become irrelevant. See [28] for a broader discussion of removing information from a knowledge base.

Finally, more sophisticated mechanisms of reinforcement learning could be implemented using custom sensors/actuators.

ACKNOWLEDGMENT

Horacio Paggi wish to thank Prof. Artem Katasonov and Prof. Vagan Terziyan for answering gently the questions related to S-APL.

REFERENCES

[1] J. Searle, Speech Acts. Cambridge University Press 1969.

[2] J. L. Austin, How to Do Things With Words. 2nd. ed. Harvard University Press, 1962.

[3] P. Sutton, "Vagueness, Communication, and Semantic Information", King's College: London, England, 2013.

[4] J. L. Austin, Sense and Sensibilia. Oxford University Press, 1962.

[5] V. Novák, "Are fuzzy sets a reasonable tool for modeling vague phenomena? ". Are fuzzy sets a reasonable tool for modeling vague phenomena? Fuzzy Sets and Systems, 2005. **153**(3): pp. 341-348.

[6] N. J. J. Smith, "Vagueness, Uncertainty and Degrees of Belief: Two Kinds of Indeterminacy; One Kind of Credence", in Prague International Colloquium on Epistemic Aspects of Many-valued Logic: Prague, Czech Republic, 2011. pp. n/a.

[7] L. Zadeh, " Outline of a new approach to the analysis of complex systems and decision processes". Outline of a new approach to the analysis of complex systems and decision processes IEEE Trans. SMC,, 1973. **3**(1): pp. 28-44.

[8] L. Zadeh, " A fuzzy-algorithmic approach to the definition of complex or imprecise concepts". A fuzzy-algorithmic approach to the definition of complex or imprecise concepts Int. Jour. Man-Machine Studies 8:249–291, 1976], 1976(8): pp. 249-291.

[9] K. Todorov, P. Geibel, and C. Hudelot, "A Framework for a Fuzzy Matching between Multiple Domain Ontologies", in Lecture Notes in Computer Science Springer, Editor, 2011 pp. pp 538-547.

[10] V. Mascardi, D. Ancona, R. H. Bordini, and A. Ricci. "CooL-AgentSpeak: Enhancing AgentSpeak-DL Agents with Plan Exchange and Ontology Services". in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. 2011.

[11] Sally M. El-Ghamrawy, Ali I. EI-Desouky, and M. Sherief. "Dynamic ontology mapping for communication in distributed multi-agent intelligent system". in International Conference on Networking and Media Convergence - ICNM 2009. IEEE.

[12] D. Ancona, V. Mascardi, J. F. Hubner, and R. H. Bordini, "Coo-AgentSpeak: Cooperation in AgentSpeak through Plan Exchange", in Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2 IEEE Computer Society: New York, New York, 2004. pp. 696-705.

[13] F. f. I. P. A. (FIPA), "FIPA Ontology Service Specification": Geneve, Switzerland, 2010.

[14] M. Cochez and M. Z. Asghar, " Multi-channel Communication using Ontology Alignment and Semantic Templates. ", in Frontiers of Information Technology 2014.

[15] A. Katasonov and V. Terziyan, Using Semantic Technology to Enable Behavioural Coordination of Heterogeneous Systems. Semantic Web, ed. G. Wu, 2010.

[16] R. H. Bordini, J. F. Hübner, and M. Wooldridge, Programming Multi-agent Systems in AgentSpeak Using Jason. Wiley Series in Agent Technology John Wiley & Sons, 2007.

[17] Vincent Louis and T. Martinez. "JADE Semantics Add-on". in AAMAS'05. 2005. Utrecht, the Netherlands.

[18] V. Pautret, "Jade Semantics Add-on Programmer's guide", France Telecom: Lannion Cedex, France, 2006.

[19] G. De Giacomo, Y. Lespérance, H. J. Levesque, and S. Sardina, "IndiGolog: A High-Level Programming Language for Embedded Reasoning Agents", in Multi-Agent Programming: Languages, Platforms and Applications 2009.

[20] S. Turnbull. "Why unitary boards are not best practice: The case for compound boards". in The First European Conference on Corporate Governance. 2000. Brussels, Belguim.

[21] Q. Pu, O. Lin, and S. Fu, "Adopting semantic language in agent communication processes", in 10th IEEE International Conference on Cognitive Informatics and Cognitive Computing IEEE: Banff, Alberta, Canada., 2011.

[22] D. G. Silva and J. C. Gluz, "AgentSpeak(PL): A New Programming Language for BDI Agents with Integrated Bayesian Network Model", in 2011 International Conference on Information Science and Applications (ICISA),, IEEE, Editor, 2011.

[23] J. C. Gluz, R. M. Viccari, C. D. Flores, and L. Seixas. "Formal analysis of a probabilistic knowledge communication framework". in 10th Ibero-American Conference on AI IBERAMIA-SBIA'06 2006. Ribeirão Preto, Brazil: Springer-Verlag Berlin, Heidelberg.

[24] K. Devlin, "Situation theory and situation semantics", in Handbook of the History of Logic, D.M. Gabbay and J. Woods, Editors., Springer, 2006.

[25] M. Cochez, "Semantic agent programming language: use and formalization", in Department of Mathematical Information Technology University of Jyväskylä: Jyväskylä, Finland, 2012.

[26] A. Katasonov, "Ubiware platform and Semantic Agent Programming Language (S-APL). Developer's guide", University of Jyvaskyla, 2009.

[27] A. Katasonov and M. Cochez, "*UBIWARE Platform, Application Developer's guide, RAB overview*", in Industrial Ontologies Group, University of Jyväskylä.: Finland, 2012.

[28] V. Ermolayev, R. Akerkar, V. Terziyan, and M. Cochez, "Toward evolving knowledge ecosystems for big data understanding. ", in Big Data Computing R. Akerkar, Editor Taylor & Francis.: Boca Raton, FL. USA, 2014 pp. 3-56.