

**This is an electronic reprint of the original article.
This reprint *may differ* from the original in pagination and typographic detail.**

Author(s): Paggi, Horacio; Cochez, Michael

Title: Indeterminacy Reduction in Agent Communication Using a Semantic Language

Year: 2015

Version:

Please cite the original version:

Paggi, H., & Cochez, M. (2015). Indeterminacy Reduction in Agent Communication Using a Semantic Language. *WSEAS Transactions on Systems*, 14, 77-89.
<http://www.wseas.org/multimedia/journals/systems/2015/a155702-568.pdf>

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Indeterminacy Reduction in Agent Communication Using a Semantic Language

HORACIO PAGGI

Engineering Faculty – LISI Group
ORT University

Mercedes y Cuareim, 11800. Montevideo

URUGUAY

horacio.paggi@gmail.com

MICHAEL COCHEZ

Department of Mathematical Information
Technology

University of Jyväskylä,

FINLAND

michael.cochez@jyu.fi

Abstract: - In recent years, the importance of vagueness and uncertainty in the messages exchanged between agents has been highlighted, mainly due to the ubiquitous nature of the (artificial or human) agents' communication. The imprecision in the communication becomes more significant when the autonomy of the agents increases or the number of exchanged messages for a communicative goal is limited. In this paper we conjugate ideas drawn from situation semantics theory, human communication, and the multi-agent systems (MAS) field to reduce the impact of vagueness and uncertainty present in the communication. The main advances are achieved with the help of context information, collaboration and reinforcement learning using an agent communication language: the Semantic Agent Programming Language (S-APL).

Key-Words: Communication semantics, multi-agent systems programming language, message vagueness

1 Introduction

1.1 General view

When agents (human or artificial), communicate, vagueness can arise related to the perceived meaning of the message (the illocution, i.e., its semantics). This, in turn, implies a variable response of the receiver (the perlocution), as is described in speech act theory [1, 2]. These variable responses can be seen as uncertainties related to the results of the actions triggered by the message in the receiver agent. These uncertainties are connected to the dynamic behavior of the system {sender, receiver} (e.g., the previous performances of the sending and receiving agents).

In some hierarchical systems, more specifically, holonic systems, it's fundamental to respond to a message coming from a higher level "in the best way possible". This is due to the semi-autonomous nature of each level's components and the fact that the number of exchanged messages should be kept to a minimum [3]. To attain this goal, the system should minimize the sending and receiving of clarifying messages. Additionally, a low number of exchanged messages is desirable when the system incorporates mobile devices with a limited energy supply and robots build on them (see for

example [30]), when there is an unreliable network connectivity (in the sense that even though the communication channel is noiseless, when we get a message, we know it is correct, but this does not mean that we are always able to send messages nor that this will be fast.) or when humans play a part in the system. This because they (we) are more prone to information overload than computers and more expensive resources [4].

The overall idea of this work is to show how we can reduce the negative effects of uncertainty and vagueness using a specific semantic language, S-APL, which allows porting ideas from the philosophy of language field into the MAS communication. This reduction is achieved by using contexts and implementing a schema of knowledge sharing and reinforcement learning. This learning means that, with time, the interaction cycles are shorter, requiring less and less additional messages. On the other hand, the dual problem of data validity and currency is solved using an artificial neural network schema.

S-APL is a semantic agent programming language, that is, a language that can be used not only as a content language but also allows the specification of the agents' behavior. It was originally developed in the Smartsources and the Ubiware project [5]. One of the main motivations for its development was the need for a language that

allows the explicit removal of existing information in the agents and the embedding of queries and rules as normal beliefs of the agent. Additionally, given its relatively simple syntax and ample expressiveness, it could be used as a Controlled Natural Language (Controlled English) for certain communications between human agents.

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

1.2 Preliminary concepts

1.2.1 Situation semantics theory

Situation semantics theory was developed by J. Barwise and J. Perry [6] as a mathematically founded analysis of semantic issues of natural language. The theory seeks to understand linguistic utterances in terms of information conveyed [7] and can be used to analyze language from an action perspective [8]. Situation theory is the name given by Devlin and Barwise to the underlying mathematics in the analyses of natural language use. From [6] we borrow many concepts:

- *A situation is a limited part of the world, used to talk about other limited parts of the world.* The information that an agent has about a situation is just part of all the information theoretically available for the situation. The situation will determine the meaning of an utterance (message).
- *Infons: are discrete informational items related to a situation that are or are not factual for the situation.* Infons can be combined recursively using several operators (conjunction, disjunction and existential and universal quantification). The possible infons depend on the objects which are part of the situation. These objects in turn, can be arbitrarily more or less granularly grouped. As Devlin states: “the ontology of situation theory has no bottom layer, every individual or situation can be subdivided into constituents, if desired. This implies that is possible to represent and analyze a domain at any degree of granularity...” [7]. The idea of information with multiple granularity has been deeply analyzed in several fields, from granular

computing to complex systems simulation (for example in [9] a formalism for this is proposed, considering a token of a Petri net as a whole matrix of elementary tokens). For the receiver agent, perhaps the most important infon is the the indication from the sender of the original message whether the answer was understood and satisfying. In turn, for the sender, it will be the answer obtained. These infons are what Sulis and other researchers call informons [10-13].

- *Meaning*: Situation semantics distinguishes two different types of meaning: abstract and meaning-in-use. The abstract one refers to the meaning of the word/phrase/sentence in general and the meaning-in-use is its meaning as it is being used in this instance (contextualized to the situation) [7]. In this work we only consider the meaning-in-use. See section 2.3.

1.2.2 Context

For brevity, we will adopt the definition of Aboud *et al.*:

“Context is any information that can be used to characterize the situation of an entity” [14].

1.2.3 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” [15]. 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]” [16]. We are interested in the vagueness and uncertainty in the way they can affect the beliefs and responses of the agents. Given a subject (an agent), vagueness arises when it tries to categorize objects with a given property (for example, as in inductive reasoning); 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 [17]).

Vagueness can be modeled as degrees of truth and is then naturally related to the fuzzy sets theory and its concept of membership function [15, 18]; it is also associated with how a phenomenon is defined (and not with its occurrence) and is typical of natural language.

Two different kinds of vagueness can be found in agent communication:

- The proper one of the vocabularies (ontologies) used in the communication. If the ontologies allow fuzzy concepts – as in the case of f-OWL [19], then a fuzzy concept (generally related to a linguistic variable) may have different values in its membership function in the sender and in the receiver. For example, the notion of a “tall person” may be given by a membership function shaped as a right shoulder (0,170,180,*) for the sender – meaning that a person who is less than 170cm. high is not tall, a person over 180cm. high is definitely tall. On the other hand, for the receiver the function could be defined as (0,185,195,*), see Fig. 1. leading to a different interpretation. To handle this vagueness we propose the use of strategies such as considering the context of the message (for example, if Peter is 15 years old, when he speaks about an old man, ‘OLD MAN’ may mean a 30 years old person). This kind of vagueness corresponds to the U1 uncertainty type of Sutton [20]. Thus, the degree of membership of the relations between the constituents of the piece of information is context specific. In other words, it is not possible to even state the membership function without knowing how the piece of information is used.

- The second type of vagueness is the one 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 [21, 22] an explicit search of the unknown concept in a set of collaborative agents’ ontologies is triggered.

Unlike vagueness, uncertainty has an epistemic character [15] 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. Here we are interested in the analysis of the uncertainty in the light of past behaviors (considering as “behaviors” the answers that an 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 [15]. 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 rules in the decision making (reasoning) process and the facts as data stored in the same container. When needed, these data incorporate information structures that record the probabilities of an answer being the most adequate for a given 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. We’ll adopt the frequentist approach here.

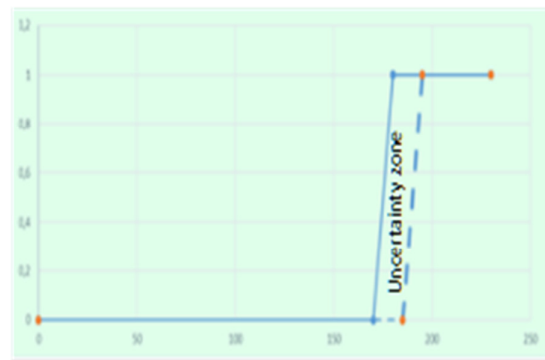


Fig. 1. U1 uncertainty type.

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 sender as well (since it, in turn, will be the receiver of the answer).

1.2.4 Neural networks and time series prediction

Neural networks have been extensively used to predict time series (see for example [23-29]). In our case, we are interested in a symbolic time series, where each symbol corresponds to a message (assuming a finite number of different messages, of course). This is a limiting assumption, necessary because the neural network has as many inputs as

symbols; nevertheless we are assuming that the number of possible messages is finite at a given instant, but it can change (grow) through time, and so will the network. Therefore we can state our problem as follows:

An *interaction* is a series of triples (w, x_i, y_i) representing the exchanged messages between agents A and B where

- w is the original message (question) received by A,
- x_i is the given answer by A, and
- y_i is the ‘quality assessment’ of x_i given by B (can be “ERROR” or “OK”).

For a given interaction $(w, x_1, y_1), \dots (x_n, y_n)$, agent A wants to answer x_n such as y_n is “OK” with the highest probability, for every n .

This is a simplified version of the communicative process: note that an additional way of minimizing the indeterminacy impact is by A asking question(s) to clarify the received message. However, the number of questions A→B and the maximum number of clarifying messages B→A (the number of clarifying cycles) has to be fixed beforehand, what is more, the definition of an interaction protocol could be needed. We would be wishing then to minimize the number of clarifying messages and questions A→B or B→A. Note that another form of ending the dialogue could be:

- a) “CANCEL” from B cancelling the request (i.e. B gives up)
- b) “UNKNOWN” if A can’t answer the message, despite the collaboration with other agents

We won’t consider these variants.

For this end, the authors of [31] use a recurrent neural network of discrete time (DTRNN) trained with the Real Time Recurrent Learning algorithm. In our case, two steps in the future should be predicted: the answer y to be given and the following message from B (“OK” is the desired). These networks have the property of smoothing exponentially the influence of the past input and outputs (see Fig. 2.). In Fig. 2 the time is represented by i , x is the input, and y the output. The dynamics of the network is:

$$y_i = f(x_i) + \mu y_{i-1} \quad |\mu| < 1 \quad (1)$$

where f is the transfer function of the neuron. The determination of μ and other parameters of the networks could be done in an adaptive way or, if a set of pairs {message, answer} were given initially (as in the case of agents instantiated from an original class) an algorithm could be used for this, as proposed in [32].

DTRNNs allow keeping memory traces, a modified version of the past, in which the past vanishes gradually (because $|\mu| < 1$), as a decaying window. This is a property shared with other network topologies having feedback memory elements, such as the time-lagged feedforward networks (TLFNs) [33].

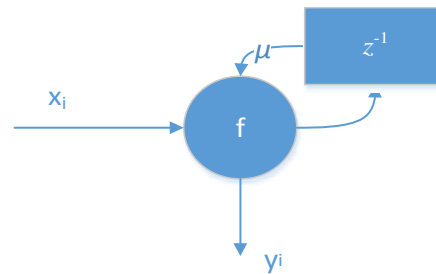


Fig. 2. The simplest DTRNN. Note that the delay operator z^{-1} is generally left implicit.

1.2.5 Holonic systems

Holonic systems are dynamic, hierarchical systems with specific properties, such as semi-autonomy of their components, efficient use of the available resources, high resiliency to disturbances (e. g. the failure or disappearance of one of their components), and adaptable to changes in the environment. The word “holon” was coined by A. Koestler referring to entities that can be considered as part of a greater one or as an individual [34]. In this sense, holonic systems are hierarchical systems in which the different elements of a level can be considered in turn as a hierarchy or as an atomic entity, depending on the analysis needs. Note that the hierarchy is evolving continuously, so a holon can be in a “higher” or “lower” level depending on the time. See [35, 36, 11, 37] for a more extended description.

Generally, holons are considered as formed by two parts: an information processing part and a physical processing part. The information processing part has a deliberative role and the other actuates on the physical environment [36].

The deliberative part can be modelled as a MAS containing abstract agents, hence the research

interests of the holonic community overlap with the ones of the MAS community.

2 Problem Formulation

In certain systems (such as the formed by semi-autonomous parts) the number of exchanged messages to attain a certain desired result of the communication has to be kept small in order to avoid an information overload of their parts. This restriction has even more impact if the system is also hierarchical, for example in the case of the holonic ones, where the mentioned number is minimum [3]. This reduced quantity of messages imply that there are very scarce chances of sending and receiving clarifying messages, so the receiver has to answer “in the best possible way” a message coming from a higher level due to the semi-autonomous nature of each level components.

Similar to Sutton [20] we won't try to define what vagueness is, because, in the words of Austin “Vague is itself vague” [38]. 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 (the one given by the situation, which supersedes the default one); 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 the system learning capability.

Note that we are considering a noiseless communication channel. In particular, 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. Hence, messages received contain no more information than is contained in the situation that originated the message [20], but we'll include the case in which information is lost, i.e. the “equivocations” of Sutton. By “information” we refer to information as content rather than information as quantity (the latter typical of Shannon's Mathematical Theory of Communication). See [20] for a deeper discussion.

The problem can then be stated as: *“how to get the receiver of a message (a holon) respond in the most desirable way (for its sender, another holon) when the received message is vague or imprecise? And how to implement this in a computational way using a limited set of tools?”*.

The constraint in the set of tools used is related to the issues of complexity and manageability that arises in the development of a system based on a vast number of components. The more tools, the

more difficult is their integration and, probably, the less computationally efficiency attained.

2.1 Alternatives for computational management of indeterminacy

2.1.1 Ontology matching

Ontologies have been suggested for use in semantic annotation and retrieval in the semantic Web, for example in [39]. The interpretation of the sent message can be seen as 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, including the case of fuzzy concepts [40]. More 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 [41]; the Coo-AgentSpeak [21] 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 [42]. 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 both ontologies are defined and maintained by another agent or organization in a third ontology, called the upper ontology (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). The relevant concepts are the concepts common to both ontologies and those which are super-classes of the original ones. In this way, when communicating actions and intentions using relevant concepts, there can be set coordinations (correspondences) between the ontologies which are evolving or that are not completely known at the given time [43].

2.1.2 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 [22]), based on AgentSpeak and its interpreter Jason [44] 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 agent shares with the former the needed answer(s), if it has any. In other words collaboration results in a plan (i.e. a series of activities needed in order to be able an answer 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 likely suitable 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 (such is the case of open MAS).

Jadex [45] 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 separate codec is needed.

Another attempt to enable the communication at a semantic level was the development of the Jade Semantic Add-on (JSA) [46, 47]. 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.

Py Ouyan and Fu [48] 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.

Silva and Gluz [49] 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 [50]. 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.

The use of Controlled Natural Languages (CNL) has been proposed to reduce or eliminate ambiguity

in the communications, basically in the inter-human ones [51]. CNLs are subsets of natural languages, obtained by restraining their grammar and vocabulary, for example the Controlled English was developed by Preece et al. as a language readable by English speakers in which information is structured and has an unambiguous form [51, 52]. S-APL could be used as a CNL provided a proper (fixed for a domain) ontology is defined.

Finally, another completely different approach is adopted in the resolution of the problem by the Information Fusion community. In that case, the indeterminacy is alleviated by the fusion of the received information with complementary data from other sources. See [53-55].

2.2 Learning, evolution and forgiveness

Given that agents learn from dialogues, it seems natural to ask when the information acquired stops being relevant. Information obsolescence is not a minor issue, citing Ermolayev et al. [56]:

“Practices in Big Data management confirm that forgetting, following straightforward policies like fixed lifetime for keeping records causes regrets almost inevitably”.

In our case, the use of a neural network with recurrent elements (ordinary recurrent networks or TLFNs) avoids the retention of useless information by summarizing the past and putting more stress in the recent history, diminishing data importance gracefully through time. Note that here agents evolve too (through learning) although it is not mandatory to use an ontology, as Ermolayev et al. do [56].

2.3 Semantics as a learned probabilistic correlation

In situation theory, meaning can be considered as a relation between the speaker connection, a context and a described situation [7]. The speaker connection is the connection (association) made between the utterance and the different objects which can be referred 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” [20]. The described situation is, in a nutshell, what is said, a situation in which the world is in some way.

An *iterated learning model (ILM)* [20] 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 using frequency recording of message patterns (message classes): the receiver learns all the possible connections of patterns and assigns a frequency (a probability) of appearance to them. When a yet unknown message arrives, the agent just has to find its class in order to have the class of the answer, which is equivalent to giving (teaching) it the correspondence between the receiver and sender classes (ontologies, if exist) and a set of classes of reference. Note that these correspondences are probabilistic.

3 Problem Solution

3.1 Overview

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

The part of the problem referring to a small set of tools is solved with S-APL because it can be used as context and behavior specification language. Features such as the reusability of behaviors and externalization of beliefs contribute to the simplicity of the implementation.

About the first part of the problem, when an agent doesn’t know the correct answer to a message, the Ubiware platform [57] (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. 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. 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.

In general, context can be:

- temporal (the sequence of received messages and answers given). The temporal context can be handled using specific techniques, for example, specialized neural networks (TLFNs, recurrent)
- information that appears in the “context field” of the message)
- sender/receiver status (their beliefs, desires and intentions).

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 [20], 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.

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

```
{
{:John :hasHeight ?x} :accordingTo
:Z.
{:John :hasHeight ?x} :accordingTo
:Q}
} sapl:implies {... .. ..}
```

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. Another option could be to retry answers which were not successful once in a while since the environment might have changed and the answer that used to be wrong might become right.

After the initial sender has sent a confirmation of a correct answer or notified about the achievement of expected results of actions, an increment in the successes count could be triggered 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.

More generally, we need to specify a procedure which registers a success whenever a message ‘OK’ is received. For example, suppose that one has a program NN that implements the on line learning of sequences of pairs

{message,answer}

(such as a neural network of an adequate type, as discussed) which takes as parameter x_n, y_n (or, equivalently, $w, (x_1, y_1), \dots, (x_n, y_n)$).

If agent A receives a message x_n from B (w for the first time), answers with y_n and B says ‘OK’ to this answer, we want to record the answer y_n and to be able to learn the sequence $w, (x_1, y_1), \dots, (x_n, y_n)$. The neural network gives an approximation of the probability of every possible message (possible symbol), so it generalizes the frequency recording. This can be done using a behavioral rule and it must be in the sapl:Rule context, as in

```
{ { condition as triples } sapl:implies {
outcome } } sapl:is sapl:Rule
```

The execution of the NN (NN for neural network) program means the execution of an action outside the beliefs of the agent, i.e., the execution of a reusable atomic behavior. This would be done like this:

```
{sapl:I      sapl:do      java:someclass}
sapl:configuredAs {parameter1  sapl:is
value1 . parameter2 sapl:is value2 . ...}
```

where *someclass* is a Java class interacting with the NN mentioned.

Given that S-APL is Turing complete it is also possible to emulate the neural network inside the agents beliefs. However, keeping the neural network external, as we showed seems most reasonable, because it allows us to reuse existing implementations and avoids the overhead of the agent’s reasoning.

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 a table would be in an external storage and could be accessed using external behaviors (java code) to find the most frequent valid answer for the present message given the previous experiences.

Note that the checking of the answer adequacy 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”.

About how to measure the reduction of the impact, three indices appear naturally:

- In the case of using a more elaborate protocol, number of additional (clarifying) messages needed to get a final answer ‘OK’ from the originating agent
- Number of times that a ‘OK’ answer is obtained in the first time, without needing any additional message
- If a cost is associated to every action of the agents, the index could be the cost of the failed actions (in the sense of not obtaining an immediate ‘OK’, or an ‘OK’ in the next two messages, etc.) for the message.

3.2 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:

- an interface agent which communicates with the user,
- a “solver” which searches actions intended to solve the problem and
- 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 a (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 rule 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 in case the solution found is accepted. 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 (see Fig. 3).

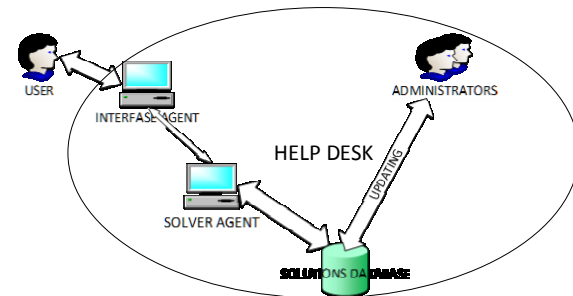


Fig. 3. Schema of the help desk

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, for instance, contain the following information:

```
: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}

```

As said, an English auction could have taken place to determine the answer A should give to a message from B, using as “price” the reputation of the other agents. Suppose that the reputation of agent C as perceived by agent D is the number of times that C gave an answer that resulted in a successful interaction for D or any other agent (this is, the getting of an ‘OK’). The code to implement such auction and to record the trustworthiness of the winner of the auction upon the reception of the confirmation from the original agent (B) is conceptually simple but lengthy since one has to model the whole communication as well. Hence, for the sake of brevity, we left out the auction itself.

4 Conclusions and future work

In this work we have sketched how communication indeterminacy can be handled (and its effects

alleviated) using an iterated learning model (ILM) and how it can be implemented in the case of software agents using the semantic agent programming language: S-APL. In the case of human agents communication exchange, its simple, orthogonal syntax and potential expressiveness lead us to think in its use as a controlled natural language, provided the needed ontology is defined. In this latter case, indeterminacy reduction can be highly desirable, for example in the HUMINT operations (see [58, 59]).

For computational agents S-APL proved beneficial for several reasons. First, the design of the agents was simplified by mixing the facts and behavioral rules. Second, the use of contexts and external belief structures allowed a hierarchical design which could reuse existing software. Finally, it was fairly easy to design the reinforcement learning loop because of the fact that the rules follow the same structure as the data.—Next, we discussed that answers given in the far past might not be as relevant as fresh ones. The issue of when certain answers should be deleted because they have become irrelevant was addressed using a neural network. We give so an alternative model of decreasing influence of the past, based on the properties of exponential attenuation of the recurrent neural networks.

We also comment how vagueness and randomness relate when related for a system which relies in communications and behaviors.

Many implementation and performance related further research questions arise:

1. 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.
2. 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
3. The specification of the holonic structure using this language is left for future research
4. Finally, it seems that the underlying ontology of S-APL could be “fuzzyfied” in order to cope with vagueness. This could be done, for example, by incorporating fOWL as description language in S-APL

References:

- [1] J. Searle, Meaning, communication, and representation, in *Philosophical Grounds of Rationality: Intentions, Categories, Ends*, G. R. and W. R., Editors, 1986, Clarendon Press, pp. 209–226.
- [2] J.R. Searle, *Speech Acts*, Cambridge University Press, 1969.
- [3] S. Turnbull, Why unitary boards are not best practice: The case for compound boards, in *The First European Conference on Corporate Governance*, 2000.
- [4] J.v. Diggelen, et al., Efficient Semantic Information Exchange for Ambient Intelligence. *Comput. J.* Vol 53, No. 8, 2010, pp. 1138-1151.
- [5] M. Cochez, Semantic agent programming language: use and formalization, in *Department of Mathematical Information Technology*, 2012, University of Jyväskylä.
- [6] J. Barwise and J. Perry, *Situations and Attitudes*. David Hume Series, ed. C.f.t.S.o.L.a. Information, University of Chicago Press, 1983.
- [7] K. Devlin, Situation theory and situation semantics, in *Handbook of the History of Logic*, D. Gabbay and J. Woods, Editors, 2008, Elsevier, pp. 601-664.
- [8] K. Devlin and D. Rosenberg, Information in the study of human interaction in *Handbook of the Philosophy of Information*, J.v.B.e. al., Editor, 2008, North Holland, pp. 685-710.
- [9] A. Spiteri Staines and F. Neri, A Matrix Transition Oriented Net for Modeling Distributed Complex Computer and Communication Systems. *WSEAS Transactions on Systems*. Vol 13, No., 2014, pp. 12-22.
- [10] F. Alonso, et al., Fundamental Elements of a Software Design and Construction Theory Informons and Holons, in *International Symposium of Santa Caterina on Challenges in the Internet and Interdisciplinary Research*, 2004.
- [11] H. Paggi and F. Alonso, Uncertainty and Randomness: A Holonic Approach, in *Second International Conference on Computer Engineering and Applications*, 2010.
- [12] H. Paggi Straneo and F. Alonso Amo, A Holonic Model of System for the Resolution of Incidents in the Software Engineering Projects, in *Proceedings of the 2009 International Conference on Computer and Automation Engineering*, IEEE, Editor, 2009, IEEE Computer Society, pp. 79-86.
- [13] W. Sulis, et al., Archetypal Dynamics, in *Formal descriptions of developing systems.*, 2003, Kluwer Academic Publishers, pp. 180-227.
- [14] G.D. Aboud, et al. *Towards a Better Understanding of Context and Context-Awareness*. in *Handheld and Ubiquitous Computing 1999*, 2001, Springer.
- [15] V. Novák, Are Fuzzy Sets a Reasonable Tool for Modeling Vague Phenomena? *Fuzzy Sets and Systems*. Vol 156 . No., 2005, pp. 341-348.
- [16] 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*, 2011.
- [17] L. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. SMC.*, Vol 3, No. 1, 1973, pp. 28-44.
- [18] L. Zadeh, A fuzzy-algorithmic approach to the definition of complex or imprecise concepts. *Int. Jour. Man-Machine Studies* 8:249–291, 1976]. Vol, No. 8, 1976, pp. 249-291.
- [19] G. Stoilos, G. Stamou, and J.Z. Pan, Fuzzy Extensions of OWL: Logical Properties and Reduction to Fuzzy Description Logics. *International Journal of Approximate Reasoning*. Vol 51, No. 6, 2010, pp. 656-679.
- [20] P. Sutton, Vagueness, Communication, and Semantic Information, 2013, King's College.
- [21] D. Ancona, et al., Coo-AgentSpeak: Cooperation in AgentSpeak through Plan Exchange, in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, 2004, IEEE Computer Society, pp. 696-705.
- [22] V. Mascardi, et al. *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.

- [23] I. El-Feghi, Z. Zubia, and S. Abozgaya, Efficient Weather Forecasting using Artificial Neural Network as Function Approximator. *International Journal of Neural Networks and Advanced Applications* Vol 1, No. 1, 2014, pp. 49-55.
- [24] S. Iliev and S. Popova, Electricity Consumption Prediction System for the Public Transportation. *WSEAS Transactions on Systems*. Vol 13, No., 2014, pp. 638-643
- [25] H.-Y. Lin and K. Chen, Soft Computing Algorithms in Price of Taiwan Real Estates. *WSEAS Transactions on Systems*. Vol 10, No. 10, 2011.
- [26] L. Macku and D. Samek. *Prediction of Semi-Batch Reactor Using Multilayered Feed-Forward Neural Networks*. in *16th WSEAS conference on systems*, 2012, WSEAS Press.
- [27] H.R. Maier and G.C. Dandy, Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling & Software*. Vol 15, No. 1, 2000, pp. 101-124.
- [28] H. Paggi and F. Robledo. *A Neural Based Model For the Prediction of the Bottled Propane Gas Sales*. in *International Conference on Mathematics and Computers in Sciences and Industry*, 2014, IEEE.
- [29] M.G. Papoutsidakis and A.G. Pipe, A Type of Radial Basis Function Technique for Control and Time Series Prediction of Positioning Systems. *WSEAS Transactions on Systems*. Vol 12, No. 11, 2013.
- [30] M. Papoutsidakis, et al., Intelligent Algorithms Based on Data Processing for Modular Robotic Vehicles Control. *WSEAS Transactions on Systems*. Vol 13, No., 2014, pp. 242-251.
- [31] J. Pérez-Ortiz, J. Calera-Rubio, and M. Forcada, Online Symbolic-Sequence Prediction with Discrete-Time Recurrent Neural Networks, in *Artificial Neural Networks — ICANN 2001*, G. Dorffner, H. Bischof, and K. Hornik, Editors, 2001, Springer Berlin Heidelberg, pp. 719-724.
- [32] M. Camilleri, F. Neri, and M. Papoutsidakis, An Algorithmic Approach to Parameter Selection in Machine Learning using Meta-Optimization Techniques. *WSEAS Transactions on Systems*. Vol 13, No., 2014, pp. 202-213.
- [33] J.C. Principe, N.R. Euliano, and W.C. Lefebvre, *Neural and Adaptive Systems: Fundamentals through Simulations*, John Wiley & Sons, 1999
- [34] A. Koestler, *The ghost in the machine*, Random House, 1982.
- [35] V. Botti and A. Giret, *ANEMONA: A Multi-agent Methodology for Holonic Manufacturing Systems*. Advanced Manufacturing ISSN 1860-5168, ed. D.T. Pham, Springer-Verlag, 2008.
- [36] V. Mařík, M. Fletcher, and M. Pěchouček, Holons & Agents: Recent Developments and Mutual Impacts, in *Multi-Agent Systems and Applications II*, V. Mařík, et al., Editors, 2002, Springer Berlin Heidelberg, pp. 233-267.
- [37] H. Paggi, et al., On the use of holonic agents in the design of information fusion systems, in *17th International Conference on Information Fusion (FUSION)*, IEEE, Editor, 2014, IEEE.
- [38] J.L. Austin, *Sense and Sensibilia*, Oxford University Press, 1962.
- [39] T. Sjekavica, Gordan Gledec, and M. Horvat. *Multimedia annotation using Semantic Web technologies*. in *7th WSEAS European Computing Conference (ECC'13)*, 2013.
- [40] B. Xu, et al., Mapping Fuzzy Concepts Between Fuzzy Ontologies, in *Knowledge-Based Intelligent Information and Engineering Systems*, R. Khosla, R. Howlett, and L. Jain, Editors, 2005, Springer Berlin Heidelberg, pp. 199-205.
- [41] S.M. El-Ghamrawy, A.I. El-Desouky, and M. Sherief, Dynamic ontology mapping for communication in distributed multi-agent intelligent system, in *International Conference on Networking and Media Convergence*, IEEE, Editor, 2009, IEEE, pp. 103-108.
- [42] F.f.I.P.A. (FIPA), FIPA Ontology Service Specification, 2010.
- [43] A. Katasonov and V. Terziyan, Using Semantic Technology to Enable Behavioural Coordination of Heterogeneous Systems, in *Semantic Web*, G. Wu, Editor, 2010, IN-TECH Publishing, pp. 135-156.
- [44] 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.
- [45] A. Pokahr, L. Braubach, and K. Jander, Jadex: A Generic Programming Model and One-Stop-Shop Middleware for Distributed

- Systems, in *PIK - Praxis der Informationsverarbeitung und Kommunikation* 2013.
- [46] V. Louis and T. Martinez, JADE Semantics Add-on, in *Autonomous Agents and Multi-Agent Systems AAMAS'05*, 2005.
- [47] V. Pautret, Jade Semantics Add-on Programmer's guide, 2006, France Telecom.
- [48] 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*, 2011, IEEE.
- [49] D.G. Silva and J.C. Gluz, AgentSpeak(PL): A New Programming Language for BDI Agents with Integrated Bayesian Network Model, in *International Conference on Information Science and Applications (ICISA)*, IEEE, Editor, 2011.
- [50] J.C. Gluz, et al., Formal analysis of a probabilistic knowledge communication framework, in *Proceedings of the 2nd international joint conference, and Proceedings of the 10th Ibero-American Conference on AI and 18th Brazilian conference on Advances in Artificial Intelligence*, 2006, Springer-Verlag, pp. 138-148.
- [51] V. Dragos, Assessment of uncertainty in soft data: a case study, in *17th International Conference on Information Fusion, FUSION 2014 2014. IEEE 2014*, IEEE, Editor, 2014, IEEE.
- [52] A. Preece, et al., Integrating hard and soft information sources for D2D using controlled natural language, in *Information Fusion (FUSION), 2012, 15th International Conference on IEEE*, Editor, 2012, IEEE, pp. 1330-1337
- [53] D.L. Hall and S.A.H. McMullen, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, 2004.
- [54] D.A. Lambert, A blueprint for higher-level fusion systems. *Inf. Fusion*. Vol 10, No. 1, 2009, pp. 6-24.
- [55] G.L. Rogova and E. Bosse, Information quality in information fusion, in *Information Fusion (FUSION), 2010 13th Conference on*, 2010, IEEE, pp. 1-8.
- [56] V. Ermolayev, et al., Toward evolving knowledge ecosystems for big data understanding. , in *Big Data Computing R. Akerkar*, Editor, 2014, Taylor & Francis., pp. 3-56.
- [57] A. Katasonov and M. Cochez, *UBIWARE Platform, Application Developer's guide, RAB overview*, in 2012, Industrial Ontologies Group, University of Jyväskylä.
- [58] V. Dragos. *Shallow semantic analysis to estimate HUMINT correlation*. in *Information Fusion (FUSION), 2012 15th International Conference on*, 2012.
- [59] E. Grimm Arsenault, The Domestic Politics Of International Norms: Factors Affecting U.S. Compliance With The Geneva Conventions, in *Faculty of the Graduate School of Arts and Sciences*, 2010, Georgetown University, pp. 230.