Towards a Satisfactory Conversion of Messages among Agent-based Information Systems

Idoia Berges*, Jesús Bermúdez, Alfredo Goñi, Arantza Illarramendi

University of the Basque Country, UPV/EHU. Paseo Manuel de Lardizabal, 1, Donostia-San Sebastián, Spain

Abstract

Over the last years, there has been a change of perspective concerning the management of information systems, since they are no longer isolated and need to communicate with others. However, from a semantic point of view, real communication is difficult to achieve due to the heterogeneity of the systems. We present a proposal which, considering information systems are represented by software agents, provides a framework that favours a semantic communication among them, overcoming the heterogeneity of their agent communication languages. The main components of the framework are a suite of ontologies –conceptualizing communication acts– that will be used for generating the communication conversion, and an Event Calculus interpretation of the communications, which will be used for formalizing the notion of a satisfactory conversion. Moreover, we present a motivating example in order to complete the explanation of the whole picture.

Keywords:

---

*Corresponding author. Phone:+34 943 01 81 51

Email addresses: idoia.berges@ehu.es (Idoia Berges), jesus.bermudez@ehu.es (Jesús Bermúdez), alfredo@ehu.es (Alfredo Goñi), a.illarramendi@ehu.es (Arantza Illarramendi)

1. Introduction

Nowadays, the vision of isolated information systems that work on their own without cooperating with other systems is no longer realistic. Information systems which have been independently developed by different organizations need to communicate with each other in order to enhance their functionality. However, several problems must be solved before real communication is achieved. Although there are several levels where communication issues may occur, such as the transport level or the application protocol level, in this paper we focus only on how to achieve utterance communication at a semantic level.

We adopt the cooperative software agents approach as communication platform since agent technology has been proved useful for solving problems with a highly distributed nature that need flexible and adaptable solutions (e.g. (Agogino and Tumer, 2012; Sato et al, 2011)). Communication between software agents is based on the interchange of messages. When an agent from one system wants to communicate with an agent of another system, it generates a message following the rules that have been established in the former system in order to write messages. These rules imply the use of fixed structures, languages and meaning, which is useful within a certain agent system but becomes useless in the majority of situations when trying to communicate between agents from heterogeneous systems. We advocate the use of a more flexible interoperation where the interpretation of a message is made dynamically and transparently for the heterogeneous agents that want

to communicate with each other.

In the relevant literature, several works can be found that consider the problem of how to achieve a flexible communication between agents from heterogeneous systems. Details of this are presented in Section 6 but, in summary, we can say that a high number of solutions point towards the development of mechanisms that only deal with syntactical aspects of agent communication languages (ACLs) (see (Lopes and Botelho, 2005), (Suguri et al, 2002)). Furthermore, a trend of works emphasize the adequacy of communication protocols in order to restrict the potentially wild set of utterance choices. Models of human dialogues have been used to categorize some types of dialogues (Walton and Krabbe, 1995) and protocol designers have adopted such source as a basis for structuring agent communications (Maudet and Chaib-Draa, 2002). Analyses have been undertaken on desiderata for such protocols (McBurney et al, 2002). Doubtless, communication protocols are an important part of the play of software agent interactions, and they are subject to a lot of study and effort (Desai et al, 2009; Berges et al, 2011; Chopra and Singh, 2009). Nevertheless, heterogeneity of already existing information systems exhibits great differences in the structure and intended semantics of their messages. Therefore it is not guaranteed that they satisfy the basic assumptions considered in some protocol models (i.e. semantics of individual messages expected in the protocol do not match with the message semantics of the participating agent). In our opinion, semantic technologies such as ontologies can be used in order to handle semantic aspects properly, and in particular, we advocate for managing semantics of individual messages in order to increase the possibilities of success when deal-

ing with protocols. Other authors share the opinion that the introduction of semantic-based technology in information systems provides for a significant enhancement of their functionalities and capabilities. For example authors in (Shaw et al, 2012) study the application of ontologies in order to create FAQ auto-categorizing systems that can improve customer service in technical support centers and alike. Moreover, in (Chen et al, 2012) ontologies and the Semantic Web Rule Language (SWRL[1]) are used in order to develop a diabetes medication recommendation system. This system aims at suggesting a prescription for a patient based on knowledge about symptoms and diabetes-related drugs. Finally authors in (Belmonte et al, 2008) describe an ontology-based multiagent decision support system for bus fleet management, whose goal is to reason about traffic behaviour in the same way an expert traffic operator would. However, as opposed to our approach, only agents that use the same ACL are considered. In this sense, we present a framework consisting of a communication acts ontology, called CommOnt, a translation mediator, an Event Calculus axiomatization of the scenario and some agents in charge of managing the communication process between systems.

In summary, the main features of the proposed framework are:

- It favours a semantic communication between heterogeneous and distributed information systems using semantic technologies and therefore it increases their cooperation opportunities.

- It incorporates an ontology that describes the types of communication

---

[1] www.w3.org/Submission/SWRL/

4

acts used by software agents. This ontology allows the recognition of instances of communication acts from one system as instances of communication acts of another system.

- It also integrates domain and action ontologies with the goal of extending its applicability to a wide range of different scenarios.

- It incorporates Event Calculus sentences to represent the scenario. Model theory is used to formalize a notion of satisfactory conversion. Event Calculus reasoners allow to check proposed conversions.

- A developed prototype allows us to observe the behaviour of the framework in specific scenarios. In particular, in this paper, the feasibility of our framework is presented in a scenario where one agent of our MedicalFIPAAgents system communicates with one agent of the Aingeru system, two healthcare information systems which use different languages, structures and meanings in the composition of their messages.

The rest of the paper is organized as follows: In Section 2 the main components of the proposed framework are presented, as well as a brief description of the communication process. Section 3 explains preliminary notions about the ontologies in the framework and the dynamic interpretation of communication acts. Section 4 introduces the formalization of the notion of satisfactory conversion, while one scenario of the proposed framework at work is presented in Section 5. Related works are discussed in Section 6 and finally, we end with some conclusions in Section 7.
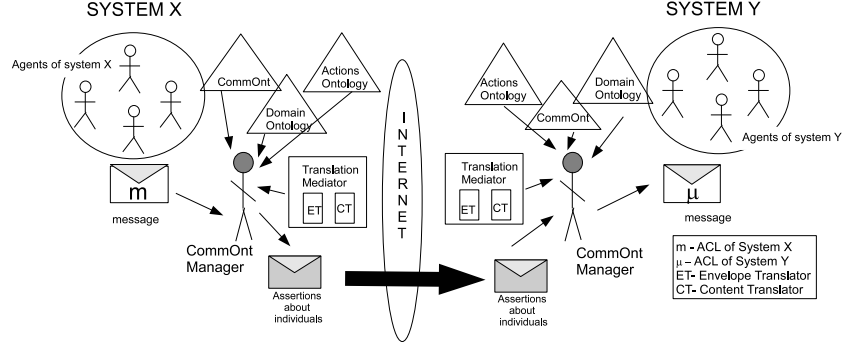
Figure 1: Architecture of the framework

2. Global architecture of the framework

In this section the architecture of the framework is presented from a global point of view. First, the elements that belong to the framework are briefly described, emphasizing the role that each element plays in the architecture. Then, the process applied to one message from the moment it is created until it reaches its final destination is shown.

2.1. Elements of the framework

In Fig. 1 the global architecture of the framework can be found. The following elements take part in it:

CommOnt and other ontologies: We have defined the CommOnt ontology to serve as a conceptualization of communication acts. Further details about this ontology will be explained in Section 3.1. Moreover, a set of ontologies formalize the conceptualization of domain actions and domain propositions.

TranslationMediator: This module is in charge of the translators that

6

transform the messages managed by the agents of each particular information system to their representation in terms of the CommOnt ontology and the other way round. In other words, translators considered in our framework produce an OWL (OWL2, 2009) document from an input message written in a given ACL and conversely, produce an ACL message given an OWL document as input.

CommOntManager: This agent is responsible for managing the process applied to a message from its creation by the sender until it reaches the receiver. In order to do this the CommOntManager needs to interact with the TranslationMediator and the ontologies. The state machine which represents the behaviour of the CommOntManagers is shown in Appendix Appendix A. We are aware that the role of the CommOntManager could be supressed from the framework, but this would imply that all the agents that belong to the information systems were able to interact with the TranslationMediators and the ontologies, so we think that it is more convenient to have a single agent specialized in that task.

2.2. Brief description of the conversion process

When an agent of information system X (see Fig. 1) wants to communicate with an agent of system Y, first it must send the message to the CommOntManager of its system. Then the following process is carried out (see Fig. 2): In the first step, the CommOntManager sends the message m to the TranslationMediator module, which translates it to the terms (i.e. classes and properties) in the CommOnt ontology. In order to do so, it first divides the message into two submessages, the envelope E(m) and the content C(m)
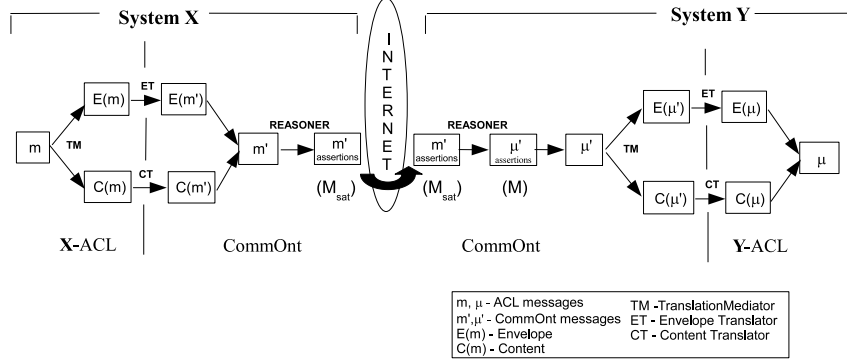
Figure 2: Conversion process applied to a message

and sends each of the submessages to its corresponding translator, ET and CT respectively. The envelope of a message contains the intention of the message; for example, to make a request or to give information about something, and also some information about the communication process, such as the identity of the sender and the receiver. The content of the message contains the object of the intention; for instance, what is the specific request made or which information is given. In general, for every ACL and for every content language there will exist a translator between those languages and COMMONT, and vice versa.

Once both submessages have been transformed to COMMONT sentences, they are re-joined. The resultant message can be considered a set of basic OWL2 assertional axioms (i.e. Description logic ABox assertions of the type $C(x)$, $R(x,y)$, where C is a class name, R is a property name, x is a named individual, and y is either a named individual or a literal value). Then, an OWL2 reasoner is launched to calculate the set of basic assertions (technically known as ABox realization) derived from the previous set of assertions in the

context of ontologies concerning the sender system X.

More formally, let M be the set of basic ABox assertions (i.e. of the form $C(x)$, $R(x,y)$ with C, R, x and y described as before) from the original message, and O^X the set of ontologies concerning the system X (including its version of CommOnt, and domain and action ontologies). Then we call M_{der} the set of new basic ABox assertions that can be derived from M in the context of O^X $((O^X, M) \vdash M_{der})$.

At this point, the saturated set $M_{sat} = M \cup M_{der}$ is materialized as OWL2 assertional axioms and sent to the CommOntManager of the information system Y. Again, an OWL2 reasoner is launched in the context of the corresponding O^Y to derive basic ABox assertions from M_{sat}. As consequence, new information is inferred, which relates the terms of the original message, that were only understood by agents of system X, with the terms in the CommOnt ontology of system Y, which are understood by the agents of system Y. Then, the new information is transformed into an OWL-format message and sent to the TranslationMediator module, which divides it into two submessages (envelope and content) and sends each one to its corresponding translator, in order to translate both parts from CommOnt to the languages used in system Y. After the translations, the resulting submessages are joined to form a complete message understood by agents of system Y. The CommOntManager can then send the message to its intended receiver, and the conversion process is finished.

Discussion

Choices made for the previously explained conversion process deserve some comments. First of all, a common formalism is selected for represent-

9

ing the diversity of ACLs. We chose the OWL2 language, which is based on a logic formalism (i.e., an expressive Description logic), and which was carefully designed for describing concepts, mantaining a trade-off between rich expressivity and decidable computational properties. OWL2 has exhibited enough expressivity in various application domains and we found it appropriate for specifying faceted classes of messages. OWL2 ontologies are logical systems that define a set of axioms that enable automated reasoning over a set of given facts. OWL2 ontology development allows a much more decentralized and collaborative approach than other non logic-based proposals. Ontology axioms corresponding to the *application layer* capture the structure and some semantics of the actual agent communication language. Application layer development is an ontology design task that must be carried out by domain experts from each information system's administration staff. Descriptions of classes and properties of messages can be checked for semantic relationships within the logic model. Translation from the actual agent communication language to the ontology language and vice versa can be made by any appropriate language translation process. We are not primarily concerned with that translation process.

Our approach presents a communication acts ontology which describes implementation independent classes and properties of messages, and which can include descriptions of established standards for ACLs. That ontology will be a *common layer* shared by all participating information systems. Ontologists who design the application layer of each information system must develop alignment axioms among classes and properties from the application layer and the common layer that will serve as an articulation ontology. Those

alignment axions are the glue of the conversion process.

An individual message from one information system will be eventually converted in a message for another information system by an inferencing process guided by the alignment axioms connecting both application layers to the articulation common layer. Some advantages of the inferencing approach against a syntax transformation approach may be considered. For instance, in case of n classes $A_1 \ldots A_n$ from the application layer such that $\forall i = 1..n \ A_i \sqsubseteq C$, for some class C in the common layer, n syntax translation processes should be defined in order to cope with any message from $A_1 \ldots A_n$, in contrast with the simple entailment step which derives $m \in C$ from $m \in A_i$ and $A_i \sqsubseteq C$. Also, a class A in the application layer may be declared as a subclass of two different classes C and D in the common layer (i.e., $A \sqsubseteq C \sqcap D$); then, using the syntax transformation approach, an *ad hoc* class should be added to the common layer in order to capture such a kind of messages, with the undesirable effect of garbage generation, in contrast with the inference based approach, which has no trouble coping with the entailments $m \in C$ and $m \in D$. Moreover, the syntax transformation approach should cope with a confluence problem in cases such as $A \sqsubseteq C$, $C \sqsubseteq D$, with A in the application layer and C, D in the common layer. Since A is subclass of C and D, two transformation process T_C and T_D must be defined; another translation process P should be defined due to $C \sqsubseteq D$, could it be guaranteed that composition of T_C with P is equivalent to T_D? In summary, the inference process approach exhibits better performance due to the underlying logic properties.

Another interesting aspect of the conversion process we want to stress

is the need for the materialization of the entailed basic assertions. It is important for the process to make explicit those assertions that describe properties of the original message with terms from the common layer; since, once sent to the target information system, they will be in charge of conveying the semantics (notice that assertions with particular application layer terms will not be understood in the recipient information system).

3. Preliminaries

In this section preliminary notions about the ontologies in the framework and our adopted approach for the dynamic interpretation of communication acts are explained.

3.1. Ontologies

As stated in the previous subsection, the developed framework makes use of different kinds of ontologies. Domain ontologies are needed for describing classes that refer to entities, properties or facts from specific knowledge areas. The terms of the domain ontologies are used, among others, in the content of messages. In Fig. 3a a subset of the domain ontology used in the AINGERU system can be found. Action ontologies are useful for describing actions that agents can perform. Classes such as `Reading`, `Writing`, `Dump`, etc. can be found in such ontologies. As it happens with domain ontologies, the terms from the action ontologies are used in the content of messages. In Fig. 3b a subset of the well known SUMO (SUMO, 2008) ontology is shown. Although domain and action ontologies are also important, the distinguishing ontology of our framework is a formal specification of a conceptualization of communication acts, which we have named CommOnt.
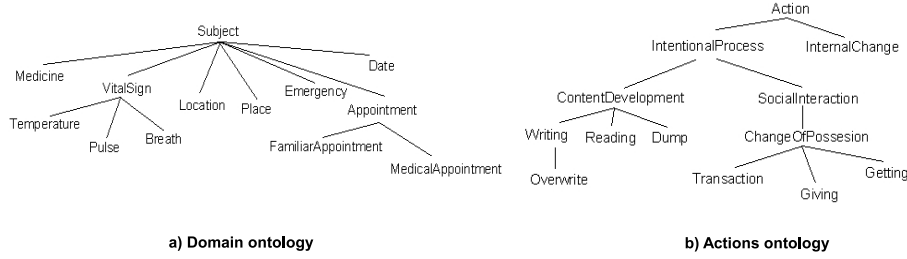
Figure 3: Domain and action ontologies

The main goal of the CommOnt ontology is to facilitate semantic communication between heterogeneous agent-based information systems. The core of the ontology is formed by the so called communication acts. The main design criteria adopted for describing communication acts has been to follow the *speech acts* theory originated by Austin ((Austin, 1962)), further developed by Searle ((Searle, 1969)) and elaborated by Vanderveken ((Vanderveken, 1990)). The speech acts theory is a linguistic theory that has been recognized as the principal source of inspiration for designing most well known standard agent communication languages. In the following section we explain briefly the specification of CommOnt, preceded by the conceptualization principles that drive that specification.

3.1.1. CommOnt ontology: Conceptualization of communication acts

According to Austin's theory every communication act may be viewed as the sender's expression of an attitude toward some possibly complex proposition. A sender performs a communication act which is expressed by a coded message and is directed to a receiver. Therefore, a communication act has two main components. First, the attitude of the sender which is called the

13

illocutionary force (F), that expresses social interactions such as informing, requesting or promising, among others (those appear in the envelope part of the message, as we mentioned in Section 2.2). And second, the *propositional content* (p) which is the subject of what the attitude is about. That perspective has been called the *F(p) framework*.

According to Vanderveken, the notion of illocutionary force is not taken as a primitive notion, but can be composed of six dimensions: an illocutionary point, a mode of achievement, propositional content conditions, preparatory conditions, sincerity conditions and a degree of strength. There are five basic illocutionary points, and for each of them there is a counterpart primitive illocutionary force which has that point: *Assertive, Commissive, Directive, Declarative,* and *Expressive*, respectively. The common features of those primitive forces are that they have one illocutionary point, no special mode of achievement of that point and a neutral degree of strength. The remaining illocutionary forces are derived from the five primitive by applying a finite number of operations: the addition of a certain propositional content condition α_1 (indicated as ϕ_{α_1}), the addition of a certain preparatory condition α_2 (σ_{α_2}), the addition of a certain sincerity condition α_3 (ψ_{α_3}), the restriction of the mode of achievement α_4 (μ_{α_4}) and increasing ($+_n$, $n \geq 1$) or decreasing ($-_n$, $n \geq 1$) the degree of strength.

For example, departing from the primitive directive force *Directive*, the following forces can be obtained:

- *Request* $= \mu_\alpha(Directive)$, where $\alpha=$"polite".

- *Urge* $= (+_1 \circ \sigma_\alpha)(Request)$, where $\alpha=$"The speaker has reasons for that course of actions"

The aforementioned operations can be composed functionally, and moreover, their composition is commutative ($f \circ g = g \circ f$).

3.1.2. COMMONT ontology: Specification

In COMMONT the $F(p)$ framework is used and different kinds of illocutionary forces and contents leading to different classes of communication acts are supported. COMMONT contains one main category to represent the communication acts. There are also two other categories to represent the actors that take part in the communication and the content of the messages, in which classes from domain and action ontologies are integrated. The terms of the COMMONT ontology are described in the form of classes and properties using the Web Ontology Language OWL2. COMMONT is composed of two interrelated layers (common layer and applications layer) that classify the communication acts regarding different levels of abstraction, being the common layer the most general and the applications layer the most specific. This division into layers allows a clearer visualization of the ontology, but it does not imply a technical division of the ontology. Every information system will have its own version of the COMMONT ontology. The common layer will be the same in the COMMONT ontology of all the systems. However, the applications layer will be tailored to each system.

The *common layer* of the COMMONT ontology includes the top class `CommunicationAct`, which represents the universal class of messages[2]. Every concrete instance of a message is an individual of this class. In addi-

---

[2]For the presentation we prefer a logic notation instead of the more verbose RDF/XML syntax.

Common Layer

a) $\quad$ CommunicationAct $\sqsubseteq$ =1 hasSender.Actor $\sqcap$ $\forall$hasReceiver.Actor $\sqcap$
$$\forall\text{hasContent.Content}$$

b) $\quad$ Directive $\sqsubseteq$ CommunicationAct $\sqcap$ $\exists$hasContent.Action

Inquiry $\equiv$ Directive $\sqcap$ $\exists$hasContent.ReportAct

c) $\quad$ Inquiry $\sqsubseteq$ $\exists$hasModeOfAchiv.{polite}

Require $\equiv$ Directive $\sqcap$ $\exists$hasContent.Command $\sqcap$
$$\exists\text{hasModeOfAchiv.\{peremptory\}} \sqcap \exists\text{hasPrepCond.\{compulsory\}}$$

FIPA-Query-Ref $\equiv$ Inquiry $\sqcap$ =1 hasContent $\sqcap$ $\forall$hasContent.FIPA-Inform-Ref

d) $\quad$ FIPA-Inform-Ref $\equiv$ ReportAct $\sqcap$ $\exists$hasQuery.RefExpression

KQML-Ask-If $\sqsubseteq$ Inquiry $\sqcap$ $\forall$hasContent.($\forall$hasQuery.Proposition)

e) $\quad$ KQML-Ask-If $\equiv$ FIPA-Query-If

Applications Layer

f) $\quad$ A-LocationQueryRef $\equiv$ Inquiry $\sqcap$ =1 hasContent.LocationInfGive
$$\sqcap\forall\text{inReplyTo.A-LocationQueryRef}$$

Figure 4: Some axioms of the CommOnt ontology

16

tion, five general subclasses of `CommunicationAct` are defined, each of which refers to a primitive illocutionary point: *Assertive*, *Directive*, *Commissive*, *Expressive* and *Declarative* (Fig. 4b). Finally, subclasses of those five primitive communication acts have been defined (Fig. 4c). Due to Vanderveken's conceptualization, membership to the primitive classes is not inferred, but asserted (that is the reason why subclass axioms are used for their specification). However, their subclasses can be specified by necessary and sufficient conditions, so that their membership can be inferred based on properties of the message.

Moreover, the common layer is extended with specific classes of messages that belong to different Agent Communication Languages (notice that different ACL standards use distinct classes, Fig. 4d). More precisely, communication acts of the standards FIPA ((FIPA-ACL, 2005)) and KQML ((KQML, 1993)) are considered. Also, alignment axioms between communication acts from different standard languages are defined in this level, so that communication between software agents that use different standard languages is feasible (Fig. 4e).

The *applications layer* is the system specific part, where communication acts that are particular of each information system are described. These communication acts constitute the particular agent communication language of a system. Moreover, they are defined as subclasses of the communication acts in the common layer of the COMMONT ontology. For example, one of the classes that belongs to the applications layer of the COMMONT ontology of the AINGERU system is the one in Fig. 4f.

In order for an information system to benefit from the interoperability

that is provided by the proposed framework, its administrator must make the effort of defining the communication acts used by that system in the applications layer of its version of the CommOnt ontology. However, notice that in case that the system is adhered to one of the described standards, there is no need to redefine those communication acts in the applications layer. Moreover, it is obvious that the more alignment axioms are defined with the terms of the common layer, the easier it will be to achieve communication between different information systems.

3.2. Dynamic interpretation of communication acts

In addition to the static representation of the communication acts in an ontology, we have defined a dynamic interpretation of them. More precisely, we have adopted a first-order logic interpretation of messages, and then, model-theory serves as the foundation of that notion. Since messages are taken as actions (i.e. communication acts), they can be distinguised by their effects, which allows us to talk about a dynamic interpretation of communication acts. The following subsections present briefly the logical foundations of our approach.

3.2.1. Event Calculus

The Event Calculus (Shanahan, 1999) offers a language (based on first-order logic) to reason about actions and their effects. Its basic ontology comprises *events*, *fluents* and *timepoints*: *events* correspond to *actions* in our context; *fluents* are predicates whose truth value may change over time. The Event Calculus contains several predicates about events, fluents and timepoints which are summarised in Table 1. See (Mueller, 2006) for the

Predicate	Textual description
Happens(e,t)	Event e happens or occurs at timepoint t.
HoldsAt(f,t)	Fluent f is true at timepoint t.
ReleasedAt(f,t)	Fluent f is released from the commonsense law of inertia at timepoint t.
Initiates(e,f,t)	Event e initiates fluent f at timepoint t.
Terminates(e,f,t)	Event e terminates fluent f at timepoint t.
Releases(e,f,t)	Event e releases fluent f at timepoint t.

Table 1: Some predicates of the Event Calculus

complete set of predicates.

The Event Calculus uses linear time. In (Mueller, 2006) a Discrete Event Calculus axiomatization (`DEC`) of its featuring predicates can be found, which restricts the timepoint sort to discrete time steps (which is adequate for our context). Table 2 shows some of the axioms in `DEC` (see the aforementioned reference for the complete set).

The Event Calculus was initially developed for commonsense reasoning. A suitable set of dedicated formulas must be asserted in order to axiomatize the domain of interest. Such formulas can be grouped into intuitive collections (See (Mueller, 2006) for additional explanations):

- *Observations* [Γ] are formulas of the form *HoldsAt(f,t)* and *ReleasedAt(f,t)*.

- A *narrative* [Δ] is a set of event ocurrence formulas and temporal ordering formulas. *Happens(e,t)* is an *event ocurrence formula*.

Axiom and textual description	Tag
$HoldsAt(f,t+1) \leftarrow HoldsAt(f,t) \wedge \neg ReleasedAt(f,t+1) \wedge$ $\qquad \neg\exists e \ (Happens(e,t) \wedge Terminates(e,f,t))$ If a fluent is true at timepoint t, the fluent is not released from the commonsense law of inertia at timepoint $t+1$, and the fluent is not terminated by any event that occurs at t, then the fluent is true at $t+1$.	(DEC5)
$HoldsAt(f,t+1) \leftarrow Happens(e,t) \wedge Initiates(e,f,t)$ If a fluent is initiated by some event that occurs at timepoint t, then the fluent is true at $t+1$.	(DEC9)
$\neg HoldsAt(f,t+1) \leftarrow Happens(e,t) \wedge Terminates(e,f,t)$ If a fluent is terminated by some event that occurs at timepoint t, then the fluent is false at $t+1$.	(DEC10)

Table 2: Subset of axioms of the Discrete Event Calculus that depict the influence of events on fluents.

- *Effect axioms* [Σ] are a set of formulas of the form $\gamma \Rightarrow Initiates(e,f,t)$ or $\gamma \Rightarrow Terminates(e,f,t)$, where γ is a context condition.

- *State constraints and Event ocurrence constraints* [Ψ] are a set of formulas of the form γ_1, $\gamma_1 \Rightarrow \gamma_2$ or $\gamma_1 \Leftrightarrow \gamma_2$, and $Happens(e_1,t) \wedge \gamma_1 \Rightarrow (\neg) Happens(e_2,t)$, respectively, where γ_1 and γ_2 are context conditions.

Finally, it is worth mentioning that by default, no unexpected effects or events occur. In the Event Calculus this is ensured by using a technique called Circumscription (McCarthy, 1980), which minimizes the extension of the predicates of the Event Calculus. The way circumscription (CIRC) is computed is out of the scope of this paper, but again the technical details can be found in (Mueller, 2006).

There are Event Calculus reasoners that are able to compute the set of observations that are true in timepoint $t+1$, given a set of observations Γ_t in timepoint t, a narrative Δ, effect axioms Σ and state and event occurrence constraints Ψ. We express this with the following notation:

DEC $\wedge$ CIRC(Σ; *Initiates, Terminates, Releases*) $\wedge$ CIRC(Δ; *Happens*) $\wedge$ $\Gamma_t \wedge \Delta \wedge \Sigma \wedge \Psi \vdash_{+1} \Phi_{t+1}(\Sigma, \Delta, \Psi, \Gamma_t)$

3.2.2. Commitments

A special kind of fluents, relevant to our context, are the so-called commitments, whose notion has been referenced often in the Artificial Intelligence literature (Singh, 1996; Castelfranchi, 1995). Different kinds of commitments have been distinguished: the psychological (or internal) commitments, the dialogical commitments, and the social commitments. Psychological commitments are established autonomously by an agent, which is committed to

his beliefs, desires and intentions only towards himself. Formal semantics based on such mental concepts has been associated to communication acts. However, that option has been criticized for its approach (Singh, 1998) as well as for its analytical difficulties (Wooldridge, 2000). On the other hand, dialogical and social commitments present an external character. Dialogical commitments stem from the work in (Hamblin, 1970) and have had a special influence in the study of agent argumentation dialogues (Walton and Krabbe, 1995); these are commitments held by participants in a dialogue to support statements they have made if questioned or challenged to do so by other participants. Social commitments (Singh, 2000; Venkatraman and Singh, 1999; Fornara and Colombetti, 2003) are established by participation in certain social situations. According to the social approach, when agents interact they become involved in social obligations to each other, which help in structuring and harmonizing multiagent systems and in attaining coherence in their actions. Those commitments are public, and therefore they are suitable for an objective and verifiable semantics of agent interaction. As a result, we have adopted this latter social approach to express the dynamic semantics of communication acts described in the CommOnt ontology.

Definition 1. *A base-level commitment $\mathsf{C}(x, y, p)$ is a ternary relation that represents a commitment made by x (the debtor) to y (the creditor) to bring about a certain proposition p.*

For example, the base-level commitment $\mathsf{C}(Alice,\ Bob,\ bidForPainting)$ indicates the commitment made by agent Alice to agent Bob to bid for a painting.

Moreover, sometimes an agent accepts a commitment only if a certain condition holds or when a certain commitment is made by another agent. This is called a conditional commitment.

Definition 2. *A conditional commitment $CC(x, y, q, p)$ is a quaternary relation that represents that if the condition q is brought out, x will be committed to y to bring about the proposition p.*

For example, CC(*Alice, Bob, pricePainting(<35M\$), bidForPainting*) indicates the commitment made by agent Alice to agent Bob to bid for a painting if the price of the painting is less than 35M\$.

Furthermore, some formulas are needed to capture the dynamics of commitments. In the following formulas (taken from (Yolum and Singh, 2004)), *e(x)* represents an event caused by *x*. The first formula declares that when a debtor of a commitment that is in force causes an event that initiates the committed proposition, the commitment ceases to hold.

FORMULA 1: *HoldsAt(C(x, y, p), t) $\wedge$ Happens(e(x), t) $\wedge$ Initiates(e(x), p, t) $\rightarrow$ Terminates(e(x),C(x, y, p), t).*

The second formula declares that a conditional commitment that is in force disappears and generates a base-level commitment when the announced condition is brought out by the creditor.

FORMULA 2: *HoldsAt(CC(x, y, c, p), t) $\wedge$ Happens(e(y), t) $\wedge$ Initiates(e(y), c, t) $\rightarrow$ Initiates(e(y),C(x, y, p), t) $\wedge$ Terminates(e(y),CC(x, y, c, p), t).*

Finally, the third formula declares that a conditional commitment disappears when the debtor brings out the committed proposition (regardless of the announced condition).

23

FORMULA 3: *HoldsAt(CC(x, y, c, p), t) ∧ Happens(e(x), t) ∧ Initiates(e(x), p, t) → Terminates(e(x),CC(x, y, c, p), t)*

4. Formalization of satisfactory conversion

Once the conversion process has finished, it is reasonable to ask whether the communicating agents agree on the understanding of the sent and received message. It is conceivable that, although their respective message classes are related properly in the communication acts ontology, their intended effects may not match accurately (i.e. the semantics of the sent message is not exactly the same as the semantics of the received message).

In this section, first, the description of the conversion context domain is summarized by presenting an excerpt of a proper axiomatization, and then our definition of a satisfactory conversion is introduced.

4.1. Description of the conversion context domain

The conversion context domain is composed of a narrative, a set of observations, a set of effect axioms and a set of state and event ocurrence constraints.

Narrative

Let *Happens(send(A(m)), t)* be a sentence for encoding the event of sending one message m, which is an instance of the communication act A, at time-point t. Since we consider only the conversion of a single message (thus, one event), our narrative is typically a singleton $\Delta_{A(m)} = \{Happens(send(A(m)), t)\}$.

Observations

Let Γ_t be a set of observations to indicate the fluents that hold at timepoint t. Some of these observations are due to the effect axioms (let us call them Γ_Σ), while others are observations to describe the features of the fluents in Γ_Σ. For example, if due to some effect axiom the observation *HoldsAt(P,t)* is obtained, where P is of class *ApplicP*, then the observation *HoldsAt(ApplicP(P),t)* indicates that at timepoint t it can be said that P is of class *ApplicP*. For a specific agent system, the classes and properties that appear in these observations belong to the applications layer of its corresponding COMMONT ontology or to the domain or action ontologies. The information including classes of the common layer (e.g. the classes of the common layer obtained by subsumption) is not considered.

Effect axioms

Formulas 1 to 3 in Section 3.2.2 are special cases of effect axioms that we group in the set Σ_C.

Another kind of effect axioms are those which describe the effects associated to the classes of communication acts. We represent them with the set Σ_T. Next we show some examples that refer to the effects of the communication acts `Inquiry` and `Responsive`, which appear in the common layer of COMMONT.

- *Initiates(send(Inquiry(s, r, P)),* CC*(r, s, accept(r, s, P), P), t)*

  By using an Inquiry, the sender expects to get an answer from the receiver to the question indicated in the content of the message. Moreover, the event of sending an Inquiry from s to r produces the effect

of generating a conditional commitment which expresses that if the receiver r accepts the demand, it will be commited to the proposition P in the content of the communication act.

- *Terminates(send(Responsive(s, r, P, RA)),*C*(s, r, RA), t).*
  *Terminates(send(Responsive(s, r, P, RA)),*CC*(s, r, accept(s, r, RA), RA), t).*
  Initiates(send(Responsive(s, r, P, RA)), P, t)
  By using a Responsive, the sender answers a previously received question or demand it committed to answer. As consequence of sending a message of the class Responsive, the commitment (either base-level or conditional) of the sender s towards the receiver r to bring about proposition RA ceases to hold, and moreover, the fluent P is initiated.

State and event occurrence constraints

Moreover, from the axioms in COMMONT, state and event ocurrence constraints are obtained. We group them in the set Ψ.

a) State constraints: Let B be a class name for representing a communication act. Let $Cexp$ be an expression of the form[3] $D \sqcap \exists p.E$, where D, E are class names and p is an object property. Then, two rules for indicating state constraints are the following:

- if $B \sqsubseteq Cexp$ then $HoldsAt(B(?m),t) \to HoldsAt(D(?m),t)$

---

[3]This is just an example of how state constraints are calculated. For other types of $Cexp$ the corresponding rules are created. The whole specification can be found in (Berges, 2011).

- if $B \sqsupseteq Cexp$ then *HoldsAt(D(?m),t)* $\wedge$ *HoldsAt(p(?m,?op))* $\wedge$ *HoldsAt(E(?op))* $\rightarrow$ *HoldsAt(B(?m))*

b) Event ocurrence constraints: Let B and C be class names for representing communication acts. Then, the following rule can be defined:

- if $B \sqsubseteq C$ then *Happens(send(B(?m)),t)* $\rightarrow$ *Happens(send(C(?m)),t)*

In conclusion, the dynamic semantics of a message is determined by the fluents that are initiated and terminated as a result of sending that message. In summary, communication acts have a dual semantic representation: The description in COMMONT of their structure and of their hierarchical relationships and then some Event Calculus formulas which specify their effects.

4.2. Definition of satisfactory conversion

In the following we present the formalization of the notion of a satisfactory conversion of a message. Let Γ_t be the set of observations that are true at time t, and let $\Sigma = \Sigma_C \cup \Sigma_T$ be a set of effect axioms. Let us assume that message $A(m)$ is sent at time t, then we call $\Phi_{t+1}(\Sigma, \Delta_{A(m)}, \Psi, \Gamma_t)$ the set of observations that are true at time $t + 1$ in the context of Σ, Ψ and Γ_t. Formally:

DEC $\wedge$ CIRC(Σ; *Initiates, Terminates, Releases*) $\wedge$ CIRC($\Delta_{A(m)}$; *Happens*) $\wedge$ $\Gamma_t \wedge \Delta_{A(m)} \wedge \Sigma \wedge \Psi \vdash_{+1} \Phi_{t+1}(\Sigma, \Delta_{A(m)}, \Psi, \Gamma_t)$

Definition 3. *Message $A_2(m)$ from system 2 in the context of Σ_2, Ψ_2 and Γ_t is a satisfactory conversion of message $A_1(m)$ from system 1 in the context of Σ_1, Ψ_1 and Γ_t if $\Phi_{t+1}(\Sigma_2, \Delta_{A_2(m)}, \Psi_2, \Gamma_t)$ is consistent and*
$\Phi_{t+1}(\Sigma_2, \Delta_{A_2(m)}, \Psi_2, \Gamma_t) \models \Phi_{t+1}(\Sigma_1, \Delta_{A_1(m)}, \Psi_1, \Gamma_t).$

Notice that $\Phi_{t+1}(\Sigma_i, \Delta_{A_i(m)}, \Psi_i, \Gamma_t)$, with both $i = 1$, 2, is computable and that Event Calculus reasoners are prepared to conduct the desired proof. Intuitively, we consider a message m' a satisfactory conversion of a message m if the effects of m' in the context of the target system are sufficient as effects of m in the context of the source system. That is to say, the intended meaning expressed by the sender (with message m) is captured by the receiver (with message m').

Discussion

One may argue that the logical consequence in Definition 3 should be satisfied in the opposite direction as well (that is, that the effects of m should also entail those of m') in order to consider that a conversion has been satisfactory. However, as stated before, our goal is to check whether the recipient system is able to understand, at least, the intended meaning of the information sent by the sender system. We consider that the recipient system may add some effects that are relevant for the pecularities of that system in such a context. Although this may strengthen meaningfully the effects of the message in that context, it still respects the effects expected by the sender system. As a result we think that placing a two-way logical consequence would be a too strong condition for communication in open environments where each agent system chooses its ACL regardless of what others do.

Moreover, some discussion may be raised due to the fact that we take into account a particular initial context Γ_t at the time of evaluating the correctness of a conversion. It is widely known that the messages exchanged between software agents are not isolated messages, but they are part of conversations regulated by communication protocols. Thus, the same message

28

can occur in the context of multiple conversations. However, we believe that it is not necessary to evaluate the correctness of a conversion for all possible conversations (that is, for every possible initial context Γ_t). If the conversion preserves enough semantics, within the context in use, then the conversion can be considered correct and can be used instead of the original message. It would be unwise to discard a satisfactory conversion done under a context Γ_t just because it is not correct for some other context Γ_t'.

For example, let us imagine two messages m_1 and m_2 which only differ in the class of their communication act: $m_1 \in A_1$ and $m_2 \in A_2$. Let m_2 be the proposed conversion for m_1 and suppose that f, g and k are fluents. Moreover, for the clarity of the example, let us assume that Ψ_1 and Ψ_2 do not play any major role in this case, so they are not described.

In the following scenario:

$$\Gamma_{t0}=\{HoldsAt(f,t0)\}$$
$$\Sigma=\Sigma_C \cup \Sigma_1 \cup \Sigma_2 \cup \Sigma_3 \text{ where:}$$
$$\Sigma_C \text{ is defined as before}$$
$$\Sigma_1= Initiates(send(A_1(?m)),k,t)$$
$$\Sigma_2=HoldsAt(f,t)\Rightarrow Initiates(send(A_1(?m)),g,t)$$
$$\Sigma_3=Initiates(send(A_2(?m)),k,t)$$
$$\Delta_{A_1(m_1)}=\{Happens(send(A_1(m_1)),t0)\}$$
$$\Delta_{A_2(m_2)}=\{Happens(send(A_2(m_2)),t0)\}$$

In order to prove that m_2 can not be considered a satisfactory conversion for m_1, first $\Phi_{t0+1}(\Sigma, \Delta_{A_1(m_1)}, \Psi_1, \Gamma_{t0})$ is calculated:

1. From Σ_1, $m_1 \in A_1$, and due to the binding of the variables taking into account the current scenario: *Initiates(send($A_1(m_1)$)),k,t0)*

2. From Γ_{t0}, $m_1 \in A_1$, and Σ_2, and due to the binding of the variables considering the current scenario: *Initiates(send($A_1(m_1)$)),g,t0)*

3. From $\Delta_{A_1(m_1)}$, 1, 2 and `DEC9` (see Table 2): *HoldsAt(k,t0+1), HoldsAt(g,t0+1)*

4. From Γ_{t0} and `DEC5`: *HoldsAt(f,t0+1)*

Thus, $\Phi_{t0+1}(\Sigma, \Delta_{A_1(m_1)}, \Psi_1, \Gamma_{t0}) = \{$*HoldsAt(k,t0+1), HoldsAt(g,t0+1), HoldsAt(f,t0+1)*$\}$.

Then, $\Phi_{t0+1}(\Sigma, \Delta_{A_2(m_2)}, \Psi_2, \Gamma_{t0})$ is calculated:

5. From Σ_3, $m_2 \in A_2$, and due to the binding of the variables taking into account the current scenario: *Initiates(send($A_2(m_2)$)),k,t0)*

6. From $\Delta_{A_2(m_2)}$, 5 and `DEC9`: *HoldsAt(k,t0+1)*

7. From Γ_{t0} and `DEC5`: *HoldsAt(f,t0+1)*

Thus, $\Phi_{t0+1}(\Sigma, \Delta_{A_2(m_2)}, \Psi_2, \Gamma_{t0})=\{$*HoldsAt(k,t0+1), HoldsAt(f,t0+1)*$\}$. In this case, it can be concluded that $\Phi_{t0+1}(\Sigma, \Delta_{A_2(m_2)}, \Psi_2, \Gamma_{t0}) \not\models \Phi_{t0+1}(\Sigma, \Delta_{A_1(m_1)}, \Psi_1, \Gamma_{t0})$.

However, if in the previous scenario the initial context Γ_{t0} is modified to $\Gamma_{t0} = \emptyset$ then fluent g is not initiated in step 2, and thus, it will not get to hold. So, in this case $\Phi_{t0+1}(\Sigma, \Delta_{A_1(m_1)}, \Psi_1, \Gamma_{t0}) = \{$*HoldsAt(k,t0+1)*$\}$, $\Phi_{t0+1}(\Sigma, \Delta_{A_2(m_2)}, \Psi_2, \Gamma_{t0}) = \{$*HoldsAt(k,t0+1)*$\}$ and as consequence: $\Phi_{t0+1}(\Sigma, \Delta_{A_2(m_2)}, \Psi_2, \Gamma_{t0}) \models \Phi_{t0+1}(\Sigma, \Delta_{A_1(m_1)}, \Psi_1, \Gamma_{t0})$, which indicates that for the context $\Gamma_{t0} = \emptyset$, m_2 is a correct conversion for m_1.

```
(Query-Ref :sender (agent-identifier :name ConditionsChecker@medicalFIPAAgents)
        : receiver (set (agent-identifier :name VitalSignAgent@aingeru))
        : content ((action (agent-identifier :name VitalSignAgent@aingeru)
                    (FIPA-Inform-Ref
                        (RefExpression
                            (VitalSign :whose (HumanAgent :name Helen))))))
)
```

Figure 5: A MEDICALFIPAAGENTS message

5. The proposed framework at work

In this section one scenario that illustrates the different steps that are followed by the proposed framework to achieve semantic communication between software agents from two different information systems is presented. System MEDICALFIPAAGENTS is composed of agents that use FIPA-ACL as agent communication language and FIPA-SL0 ((FIPA-SL, 2002)) as content language at the time of composing messages. Moreover, system AINGERU is composed of agents that use specific communication acts in their messages (described in the applications layer of the COMMONT ontology). The format of the messages understood by the agents in AINGERU is the OWL format.

In this example, the ConditionsChecker agent of system MEDICALFI-PAAGENTS wants to send the message in Fig. 5 to the VitalSignAgent in system AINGERU in order to make a request for the information concerning the vital signs of the patient Helen.

When the CommOntManager of the system MEDICALFIPAAGENTS receives the message from an agent of its own system, firstly, it transfers it to the TranslationMediator module, which decomposes it into two parts: the envelope and the content. This decomposition is performed taking into ac-

31

```
<CommOnt:FIPA-Query-Ref ID="Message01">
        <CommOnt:hasSender>
                <CommOnt:Actor ID="Actor01">
                        <CommOnt:name>ConditionsChecker@medicalFIPAAgents</CommOnt:name>
                </CommOnt:Actor>
        </CommOnt:hasSender>
        <CommOnt:hasReceiver>
                <CommOnt:Actor ID="Actor02">
                        <CommOnt:name>VitalSignAgent@aingeru</CommOnt:name>
                </CommOnt:Actor>
        </CommOnt:hasReceiver>
        <CommOnt:hasContent>
                <CommOnt:FIPA-Inform-Ref ID="FIR01">
                        <CommOnt:hasQuery>
                                <CommOnt:RefExpression ID="RE01">
                                        <CommOnt:hasSubject>
                                                <CommOnt:VitalSign ID="VS01">
                                                        <CommOnt:whose>
                                                                <CommOnt:HumanAgent ID="Actor03">
                                                                        <CommOnt:name>Helen</CommOnt:name>
                                                                </CommOnt:HumanAgent>
                                                        </CommOnt:whose>
                                                </CommOnt:VitalSign>
                                        </CommOnt:hasSubject>
                                </CommOnt:RefExpression>
                        </CommOnt:hasQuery>
                </CommOnt:FIPA-Inform-Ref>
        </CommOnt:hasContent>
</CommOnt:FIPA-Query-Ref>
```

Figure 6: Message in CommOnt

count the syntax of the message. In a FIPA-ACL message, the content is preceded by the tag ":content", so it is easy to see that the content will be the text that goes from that tag to the following FIPA-ACL tag (or the end of the message). Fig. 5 shows both parts of the decomposed message (envelope in bold).

The following step consists on translating both submessages to the terms in the COMMONT ontology. In order to do so, a FIPA-ACL→COMMONT translator will be used to translate the envelope of the message from FIPA-ACL to COMMONT, whereas a FIPA-SL0→COMMONT translator will be used to translate the content of the message from FIPA-SL0 to COMMONT. Further details of these translators can be found in (Berges, 2011).

Then, both submessages are unified to create a complete message written in COMMONT (Fig. 6) and given back to the CommOntManager. Next, the

unified message is asserted[4] into the CommOnt ontology and, with the help of a reasoner, entailments from the source message are collected (so that the information of the original message is made explicit). Some of the most relevant assertions, including entailed assertions, are shown next[5]:

⟨Message01	rdf:type	FIPA-Query-Ref⟩	⟨FIR01	rdf:type	ReportAct⟩
⟨Message01	rdf:type	Inquiry⟩	⟨FIR01	rdf:type	Content⟩
⟨Message01	rdf:type	Directive⟩	⟨FIR01	hasQuery	RE01⟩
⟨Message01	hasModeOfAchiv	'polite'⟩	⟨RE01	rdf:type	RefExpression⟩
⟨Message01	hasContent	FIR01⟩	⟨RE01	hasSubject	VS01⟩
⟨FIR01	rdf:type	FIPA-Inform-Ref⟩	⟨VS01	rdf:type	VitalSign⟩

Finally, the CommOntManager makes inquiries to the CommOntManagerAssistant[6] about the identity and location of the CommOntManager of the receiver's system (AINGERU system in this case) and sends the message to it.

The set of OWL2 assertions sent by the CommOntManager of the system MEDICALFIPAAGENTS is received by the CommOntManager of the

---

[4]In the context of Description logics, asserted temporarily in the ABox for ABox realization.

[5]For the sake of visual clarity, the namespace CommOnt is not included in the items that appear in the assertions.

[6]The CommOntManagerAssistant manages the information of existing CommOntManagers. When an information system incorporates its CommOntManager for the first time, this event must be shared with the CommOntManagers of the other systems, so that they are aware that it is possible to communicate with the new CommOntManager from that moment onward.

AINGERU system. Then, the first step consists of asserting the set of axioms into the COMMONT ontology to infer new assertions that relate the terms of the original message with the ones understood by the agents of the AINGERU system. This will be possible thanks to the alignment axioms that have been defined in the COMMONT ontology between the terms in the applications layer and the terms in the common layer.

For the sake of the example, let us imagine that the following terms are defined in the applications layer of the COMMONT ontology of the AINGERU system:

- `A-VitalSignQueryRef` $\equiv$ `Inquiry` $\sqcap$ $\exists$`hasContent.VitalSignInfGive`

- `VitalSignInfGive` $\equiv$ `ReportAct` $\sqcap$ $\exists$`hasQuery.VitalSignInfRef`

- `VitalSignInfRef` $\equiv$ `RefExpression` $\sqcap$$\exists$`hasSubject.VitalSign`

Because of the assertions ⟨`RE01 rdf:type RefExpression`⟩, ⟨`RE01 hasSubject VS01`⟩, ⟨`VS01 rdf:type VitalSign`⟩ and the definition of the class `VitalSignInfRef`, the new assertion ⟨`RE01 rdf:type VitalSignInfRef`⟩ is inferred.

Moreover, due to the assertions ⟨`FIR01 rdf:type ReportAct`⟩, ⟨`FIR01 hasQuery RE01`⟩, ⟨`RE01 rdf:type VitalSignInfRef`⟩ and the definition of the class `VitalSignInfGive`, the new assertion ⟨`FIR01 rdf:type VitalSignInfGive`⟩ is inferred.

Finally, thanks to the assertions ⟨`Message01 rdf:type Inquiry`⟩, ⟨`Message01 hasContent FIR01`⟩, ⟨`FIR01 rdf:type VitalSignInfGive`⟩ and the definition of the class `A-VitalSignQueryRef`, the new assertion ⟨`Message01 rdf:type A-VitalSignQueryRef`⟩ is inferred. The same procedure is applied to the remaining assertions until all the new information is obtained.

```
<CommOnt:A-VitalSignQueryRef ID="Message01">
        <CommOnt:hasSender>
                <CommOnt:Actor ID="Actor01">
                        <CommOnt:name>ConditionsChecker@medicalFIPAAgents</CommOnt:name>
                </CommOnt:Actor>
        </CommOnt:hasSender>
        <CommOnt:hasReceiver>
                <CommOnt:Actor ID="Actor02">
                        <CommOnt:name>VitalSignAgent@aingeru</CommOnt:name>
                </CommOnt:Actor>
        </CommOnt:hasReceiver>
        <CommOnt:hasContent>
                <CommOnt:VitalSignInfGive ID="FIR01">
                        <CommOnt:hasQuery>
                                <CommOnt:VitalSignInfRef ID="RE01">
                                        <CommOnt:hasSubject>
                                                <CommOnt:VitalSign ID="VS01">
                                                        <CommOnt:whose>
                                                                <CommOnt:HumanAgent ID="Actor03">
                                                                        <CommOnt:name>Helen</CommOnt:name>
                                                                </CommOnt:HumanAgent>
                                                        </CommOnt:whose>
                                                </CommOnt:VitalSign>
                                        </CommOnt:hasSubject>
                                </CommOnt:VitalSignInfRef>
                        </CommOnt:hasQuery>
                </CommOnt:VitalSignInfGive>
        </CommOnt:hasContent>
</CommOnt:A-VitalSignQueryRef>
```

Figure 7: Message received by the recipient agent

The next step consists of writing the information represented by the new set of assertions into an OWL format message. In Fig. 7 the generated message is shown. It can be observed that now the message contains individuals whose most direct classes (A-VitalSignQueryRef, VitalSignInfGive, VitalSignInfRef) are specific to the AINGERU system. In the last step, the message in the OWL format must be translated into a format understood by the agents of the recipient system. In order to do so, first the message must be decomposed into its two parts (the envelope and the content). In this case, this task is also an easy one because the content is located between the <CommOnt:hasContent> and </CommOnt:hasContent> tags. Then again the use of some translators is required in order to perform the translation of both the envelope and the content (notice that in our specific example, these last two steps are avoidable, because the message format understood

35

by the agents of the AINGERU system is the OWL format itself, so there is no need to use a translator). So, finally, the CommOntManager of the AINGERU system looks which specific agent of its system must be the receiver of the message, VitalSignAgent in this case, and it sends the message to it. Next, a proof which shows that the conversion is satisfactory is presented.

Proof: "Conversion of Message01 is satisfactory"

For the sake of brevity, the names that appear in the example have been modified to the following:

$$
\begin{array}{ll}
Fqr=Fipa\text{-}Query\text{-}Ref & Avsqr=A\text{-}VitalSignQueryRef \\
Fir=Fipa\text{-}Inform\text{-}Ref & Vsir=VitalSignInfRef \\
m=Message01 & Vsig=VitalSignInfGive \\
a01=Actor01 & a02=Actor02 \\
f01=FIR01 &
\end{array}
$$

We need to prove that $\Phi_{t0+1}(\Sigma,\Delta_{Avsqr(m)},\Psi_A,\Gamma_{t0}) \Rightarrow \Phi_{t0+1}(\Sigma,\Delta_{Fqr(m)},\Psi_M,\Gamma_{t0})$ is valid, where Ψ_A and Ψ_M refer to the constraints of systems AINGERU and MEDICALFIPAAGENTS respectively.

The following elements are common to both systems:

(0) $\Gamma_{t0}=\emptyset$

(1) $\Sigma=\{Initiates(send(Inquiry(?s,?r,?P)),CC(?r,?s,accept(?r,?s,?P),?P),t)\}$

For system MEDICALFIPAAGENTS we know that

(2) $\Delta_{Fqr(m)}=\{Happens(send(Fqr(m)),t0)\}$

(3) $\Psi_{Fqr_1}, \Psi_{Fqr_2} \in \Psi_M$, where:

(3.1) Ψ_{Fqr_1}=Happens(send(Fqr(?m)),t)$\rightarrow$ Happens(send(Inquiry(?m)),t)

(3.2) Ψ_{Fqr_2}=HoldsAt(ReportAct(?r),t) $\wedge$ HoldsAt(hasQuery(?r,?q),t) $\wedge$

HoldsAt(RefExpression(?q),t) $\rightarrow$ HoldsAt(Fir(?r),t)

Now $\Phi_{t0+1}(\Sigma, \Delta_{Fqr(m)}, \Psi_M, \Gamma_{t0})$ is calculated:

From (2),(3.1) we get

(4) Happens(send(Inquiry(m)),t0)

From binding variables in (1) with elements of m we get

(5) Initiates(send(Inquiry(a01,a02,f01)),CC1,t0),

where $CC1$=CC(a02,a01,accept(a02,a01,f01),f01)

From (4),(5) and `DEC9` we get

(6) HoldsAt(CC1,t0+1)

From the assertions in the source system (see Fig. 6) about the content of m
and the law of inertia we get

(7) HoldsAt(Fir(f01),t0+1)

(8) HoldsAt(hasQuery(f01,RE01),t0+1)

(9) HoldsAt(RefExpression(RE01),t0+1)

Then, $\Phi_{t0+1}(\Sigma, \Delta_{Fqr(m)}, \Psi_M, \Gamma_{t0})$ ={(6),(7),(8),(9)}

From system AINGERU we know that

(10) $\Delta_{Avsqr(m)}$=Happens(send(Avsqr(m)),t0)

(11) $\Psi_{Avsqr_1}, \Psi_{Avsqr_2}, \Psi_{Avsqr_3} \in \Psi_A$, where:

    (11.1) Ψ_{Avsqr_1}=Happens(send(Avsqr(?m)),t)→

$$Happens(send(Inquiry(?m)),t)$$

    (11.2) Ψ_{Avsqr_2}=HoldsAt(Vsig(?v),t)→HoldsAt(ReportAct(?v),t)

    (11.3) Ψ_{Avsqr_3} =HoldsAt(Vsir(?v),t)→HoldsAt(RefExpression(?v),t)

Now $\Phi_{t0+1}(\Sigma, \Delta_{Avsqr(m)}, \Psi_A, \Gamma_{t0})$ is calculated:

From (10),(11.1) we get

(12) Happens(send(Inquiry(m),t0))

From binding variables in (1) with elements of m we get

(13) Initiates(send(Inquiry(a01,a02,f01)),CC1,t0),

    where CC1=CC(a02,a01,accept(a02,a01,f01),f01)

From (12),(13) and `DEC9` we get

(14) HoldsAt(CC1,t0+1)

From the assertions in the AINGERU system (see Fig. 7) about the components of m and the law of inertia we get

(15) HoldsAt(Vsig(f01),t0+1)

(16) HoldsAt(hasQuery(f01,RE01),t0+1)

(17) *HoldsAt(Vsir(RE01),t0+1)*

Then, $\Phi_{t0+1}(\Sigma, \Delta_{Avsqr(m)}, \Psi_A, \Gamma_{t0}) = \{(14),(15),(16),(17)\}$

Once the set of observations at timepoint *t0+1* has been calculated for each system, and considering that $\{(14),(15),(16),(17)\}$ is glaringly consistent, we have to prove that $\{(14),(15),(16),(17)\} \Rightarrow \{(6),(7),(8),(9)\}$.
Notice that $(14)=(6)$ and $(16)=(8)$. From $(15),(11.2)$ we get

(18) *HoldsAt(ReportAct(f01),t0+1)*

From $(17),(11.3)$ we get

(19) *HoldsAt(RefExpression(RE01),t0+1)*, and $(19)=(9)$

From $(18),(16),(19),(3.2)$ we get

(20) *HoldsAt(Fir(f01),t0+1)*, and $(20)=(7)$

and the proof is completed.

It can be concluded that, thanks to the proposed framework, two agents from different systems, which previously could not understand each other, have been able to communicate with one another without establishing a priori agreement on the format and vocabulary used in the interchanged messages.

6. Related works

Several works that treat the topic of interoperability among agents from heterogeneous systems can be found in the specialized literature. The work

of (Lopes and Botelho, 2005) suggests the idea of translating the content of every message to an abstract logic language defined by the authors (and vice versa). This work is similar to ours because it makes use of an intermediate language. Nevertheless, it only takes into account the syntax of the messages, so it allows interoperability between messages with different formatting but does not address the problem from a semantic point of view. In our case, however, semantic interoperability is considered.

A different approach for interoperability, based on the inclusion of pre-formatted message templates within the advertised capability description of agents is presented in (Payne et al, 2002). The shallow-parsing template approach presented in that paper relaxes the constraint that states that agents need to share a common language for describing the content and format of messages. However, it reduces the problem to sharing common templates and as a side effect imposes certain restrictions on the type of messages that can be exchanged. From our point of view, a main disadvantage is its heavy emphasis on syntactic aspects.

Mechanisms that translate communications from one multiagent system to another have been developed. For instance, (Giampapa et al, 2000) describes an InterOperator that implements a connection between the RETSINA system (a KQML-based system) and the OAA system. The Open Agent Architecture (OAA) (Martin et al, 1999) is a framework for constructing multiagent systems and their designers intended to minimize the effort involved in wrapping legacy applications. Agent communications are represented as events. Each event has a type, a set of parameters, and a content. The allowable content and parameters vary according to the type of event. In

our opinion the great effort needed to implement this ad hoc interoperation undermines the scalability of the approach.

The role of the CommOnt ontology explained in this paper is similar to the role of the PSL (Process Specification Language) ontology in (Grüninger and Kopena, 2005), but we complement the system with an Event Calculus axiomatization of the framework in order to be able to sanction the satisfaction on the conversion of messages. Another work that takes into account the conversion problem is (Dou et al, 2003), where the first-order logic language Web-PDDL is used as a common representation language and a dedicated reasoner Onto-Engine is used as inference mechanism. However, the fact that OntoEngine is not a complete reasoner and that Web-PDDL is a proprietary language, make a difference with our work that use existing complete reasoners for the standard language OWL2-DL. Moreover, authors in (Pasha et al, 2006) propose the use of OWL ontologies to allow an effective and bidirectional communication between FIPA-compliant software agents and OWL-based Web Services. More precisely, they have defined a series of alignment axioms between the standard content language FIPA-SL and its representation in OWL, so that the latter can be used by Web Services. In (Pasha et al, 2010) their work is extended in order to allow negociation between agents and web services. However, this solution is centered only on one content language, so it needs to be developed more widely to achieve any real interoperability.

Another set of works deals with the semantics of agent communication languages. (Boella et al, 2009) introduces a semantics based on social networks. It proposes a semantics of speech acts as 'relationship building', where the shape of those relationships are those usually found in social networks

(e.g. dependency, authority, commitment). In other words, in this approach the pre- and post-conditions of speech acts are expressed as constraints on social networks. Authors in (Boella et al, 2006) present a role-based semantics for ACLs, understanding the notion of role as the description of an expected behaviour. The proposed approach embeds both the mental and social semantics approaches. On the one hand, the mentalistic approach is embedded by attributing mental attitudes to the public roles agents play instead of agents themselves. That is, it moves from the agents' beliefs and goals to the roles' beliefs and goals. On the other hand, the social approach is embedded by showing how roles maintain the normative character of social semantics. Finally, the work in (Gaudou et al, 2006) presents a semantics for FIPA-ACL, based on social attitudes. The private attitudes of mentalistic approaches are replaced by attitudes that are made public through communication. It also differs from the social commitments approach, since the state of the commitment of the participating agents is obtained directly from the logical post-conditions of the communication acts (i.e. publicly expressed intentions) instead of from what has been communicated and the commitments the agents have made by doing that.

Recent research (Chopra et al, 2012) has pointed out the consensus on using social abstractions for the specification of semantics for communication acts, instead of mentalist abstractions. Works such as (Fornara et al, 2007; Singh, 2000) use social commitments to express that kind of semantics. Moreover, the Event Calculus (Mueller, 2006) has been used to represent and to reason about such commitments (Yolum and Singh, 2004; Fornara and Colombetti, 2009). In our work we also deal with social commitments and the Event Cal-

culus. However, the interesting difference between those works and our is that we focus on a different problem. Those works are concerned with the clean specification of semantics for communication acts that are assumed to be shared by the agents in the conversation; however, we are concerned with the problem of facilitating the communication to agents that eventually use different communication languages that must be somehow aligned and in evaluating the comprehension of the exchanged messages.

To the best of our knowledge, there are few works considering the Semantic Web technology for the representation of agent communication. A remarkable exception is (Fornara and Colombetti, 2010) where an OWL2 ontology is developed for representing commitments and their related items. The proposed ontology is designed with the goal of exploiting OWL reasoners in the run-time monitoring of the communication exchanged in the context of artificial institutions. Such reasoning can be considered a technical alternative to the reasoning procedures offered by the Event Calculus reasoners. Nevertheless, the goal of that work is to evaluate the compliance of the communicating agents to the norms and obligations of an artificial institution; and therefore is complementary to our work in this paper that is focussed on evaluating the comprehension of the exchanged messages.

Finally, the closest work to ours is that of (Chopra and Singh, 2008), which also proposes commitment-based interoperability. In particular, they consider constitutive interoperability, which takes into account only the meaning of messages, opposed to regulative interoperability, which takes into consideration message order, occurrence and data flow. Our work considers also this high-level definition of interoperability that takes into account the busi-

ness meaning of communication, but it differs from theirs in the decision procedure for determining interoperability of pairs of agents.

7. Conclusions

We have introduced a proposal, based on Semantic Web technology, to tackle the problem of semantic communication among heterogeneous and distributed information systems represented by software agents. A novel feature of that proposal is that it favours a flexible communication process among those systems avoiding *a priori* agreements about interchanged messages. Such flexibility is obtained by considering the semantics associated to the communication process. The proposed framework includes as central element an ontology whose main category represents communication acts according to the well-recognized speech acts theory. This ontology integrates, using alignment axioms, descriptions of classes of messages from different ACL standards as well as descriptions of non standard classes of messages from particular information systems. Alignment axioms in that ontology provide the setting for the conversion of a message from one system into a message for another system. We have explained the necessary stages to be accomplished in a communication process in order to present the whole task to be solved. Finally, we have developed a first-order logic interpretation of the scenario and we have summarized the conversion context domain by presenting a guide for a proper axiomatization, which has been used to formalize the notion of satisfactory conversion of messages. The evaluation of whether a conversion has been satisfactory opens the doors to solving a bigger problem to be addressed in the future: assessing the quality of a conversion.

Under this title we refer to the idea of informing the participants about the degree of understanding that has been achieved in a communication, that is to say, if the understanding has been total or partial, and in the latter case, suggesting some possible changes to facilitate understanding. Previous works of members of our research group in the area of estimation of information loss for multiontology based query processing (Mena et al, 2000) will be a good start point to tackle the considered problem.

Acknowledgements

References

Agogino AK, Tumer K (2012) A multiagent approach to managing air traffic flow. Autonomous Agents and Multi-Agent Systems 24(1):1–25

Austin JL (ed) (1962) How to do things with words. Oxford University Press

Belmonte MV, Pérez-de-la-Cruz JT, Triguero F (2008) Ontologies and Agents for a Bus Fleet Management System. Expert Systems with Applications 34(2):1351–1365

Berges I (2011) Semantic Interoperability among Heterogeneous Information Systems: Two Use Cases. PhD Thesis, University of the Basque Country, UPV/EHU, Spain

Berges I, Bermúdez J, Goñi A, Illarramendi A (2011) A Mechanism for Discovering Semantic Relationships among Agent Communication Protocols. Autonomous Agents and Multi-Agent Systems 23(3): 453-485

Boella B, Damiano R, Hulstijn J, van der Torre LWN (2006) Role-based semantics for agent communication: embedding of the 'mental attitudes' and 'social commitments' semantics. In: 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, pp 688–690

Boella B, Hulstijn J, van der Torre LWN (2009) Social network semantics for agent communication. In: 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009(2)), Budapest, Hungary, pp 1215–1216

Castelfranchi C (1995) Commitments: From individual intentions to groups and organizations. In: Proceedings of the First International Conference on Multiagent Systems (ICMAS-95), June 12-14, 1995, San Francisco, California, USA, pp 41–48

Chen RC, Huang YH, Bau CT, Chen SM (2012) A Recommendation System based on Domain Ontology and SWRL for Anti-diabetic Drugs Selection. Expert Systems with Applications 39(4): 3995–4006

Chopra AK, Singh MP (2008) Constitutive interoperability. In: Padgham L, Parkes DC, Müller JP, Parsons S (eds) AAMAS (2), IFAAMAS, pp 797–804

Chopra AK, Singh MP (2009) Multiagent Commitment Alignment. In: 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Volume 2, Budapest, Hungary, pp 937–944

Chopra AK, Artikis A, Bentahar J, Colombetti M, Dignum F, Fornara N, Jones AJI, Singh MP, Yolum P (2012) Research Directions in Agent Communication. Transactions on Intelligent Systems and Technology, in press.

DARPA Knowledge Sharing Initiative (1993), Specification of the KQML Agent-Communication Language. http://www.cs.umbc.edu/kqml/kqmlspec/spec.html.

Desai N, Chopra AK, Singh MP (2009) Amoeba: A Methodology for Modeling and Evolving Cross-Organizational Business Processes. ACM Transactions on Software Engineering Methodology 19(2)

Dou D, McDermott DV, Qi P (2003). Ontology Translation on the Semantic Web. In: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE - OTM Confederated International Conferences, Catania , Sicily, Italy, pp 952–969

FIPA-ACL (2005) FIPA communicative act library specification. http://www.fipa.org/specs/fipa00037/SC00037J.html

FIPA-SL (2002) FIPA SL content language specification. http://www.fipa.org/specs/fipa00008/SC00008I.html

Fornara N, Colombetti M (2003) Defining interaction protocols using a commitment-based agent communication language. In: AAMAS '03: Pro-

ceedings of the second international joint conference on Autonomous agents and multiagent systems, ACM Press, New York, NY, USA, pp 520–527

Fornara N, Viganò F, Colombetti M (2007) Agent communication and artificial institutions. Autonomous Agents and Multi-Agent Systems 14(2) pp 121–142

Fornara N, Colombetti M (2009) Specifying artificial institutions in the event calculus. In Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models, V. Dignum, Ed. IGI Global, Chapter 14, 335-366.

Fornara N, Colombetti M (2010) Representation and monitoring of commitments and norms using OWL. AI Communications 24(3), pp 341–356

Francisco V, Gervas P, Peinado F (2010) Ontological reasoning for improving the treatment of emotions in text. Knowledge and Information Systems 25(3):421–443

Gaudou B, Herzig A, Longin D, Nickles M (2006) A New Semantics for the FIPA Agent Communication Language Based on Social Attitudes. In: 17th European Conference on Artificial Intelligence (ECAI 2006), Riva del Garda, Italy, pp 245–249

Genesereth MR, Fikes RE (1992) Knowledge Interchange Format Reference Manual. Computer Science Department, Stanford University, http://ksl.stanford.edu/knowledge-sharing/kif/

Giampapa JA, Paolucci M, Sycara K (2000) Agent interoperation across

multiagent system boundaries. In: Proceedings of the Fourth International Conference on Autonomous Agents, ACM Press, pp 179–186

Grüninger M, Kopena J (2005) Semantic Integration through Invariants. AI Magazine 26(1):11–20

Hamblin CL (1970) Fallacies. Methuen, London, UK

Lopes A, Botelho LM (2005) Chomsky: A content language translation agent. In: Proceedings of Multi-Agent Systems and Applications IV, 4th International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2005, Budapest, Hungary, pp 535–538

Martin DL, Cheyer AJ, Moran DB (1999) The open agent architecture: A framework for building distributed software systems. Applied Artificial Intelligence 13(1):91–128

Maudet N, Chaib-Draa B (2002) Commitment-based and dialogue-game-based protocols: new trends in agent communication languages. The Knowledge Engineering Review 17:157–179

McBurney P, Parsons S, Wooldridge M (2002) Desiderata for Agent Argumentation Protocols. In: The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, Bologna, Italy, pp 402–409

McCarthy J (1980) Circumscription - A Form of Non-Monotonic Reasoning. Artificial Intelligence 13:27–39

Mena E, Kashyap V, Illarramendi A, Sheth AP (2000) Imprecise Answers in Distributed Environments: Estimation of Information Loss for Multi-Ontology Based Query Processing. International Journal of Cooperative Information Systems 9(4):403–425

Mueller ET (2006) Commonsense Reasoning. Morgan Kauffman

OWL Working Group (2009) OWL 2 Web Ontology Language: Document Overview. W3C Recommendation, available at `http://www.w3.org/TR/owl2-overview/`

Pasha M, ur Rehman S, Ali A, Ahmad HF, Suguri H (2006) Middleware between owl and fipa ontologies in the semantic grid environment. In: Proceedings of the 2006 International Conference on Semantic Web & Web Services, SWWS 2006, Las Vegas, Nevada, USA, pp 30–35

Pasha M, Oquendo F, Ahmad HF (2010) Negotiation in Heterogeneous Environments. In: Seventh International Conference on Information Technology: New Generations, Las Vegas, Nevada, USA, pp 290–295

Payne T, Singh R, Sycara K (2002) Communicating agents in open multi-agent systems. In: First GSFC/JPL Workshop on Radical Agent Concepts (WRAC), McLean, VA, USA

Sato GY, Silva de Azevedo HJ, Barthes JPA (2012) Agent and multi-agent applications to support distributed communities of practice: a short review. Autonomous Agents and Multi-Agent Systems 25(1): 87–129

Searle JR (1969) Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, New York

Shanahan M (1999) The event calculus explained. Lecture Notes in Computer Science 1600:409–430

Shaw RS, Tsao CF, Wu PW (2012) A Study of the Application of Ontology to an FAQ Automatic Classification System. Expert Systems with Applications 39(14):11593–11606

Singh MP (1996) A Conceptual Analysis of Commitments in Multiagent Systems. Tech. Rep. Technical Report 96-09, Department of Computer Science, North Carolina State University, Raleigh, NC, USA

Singh MP (1998) Agent Communication Languages: Rethinking the Principles. IEEE Computer 31(12):40–47

Singh MP (2000) A social semantics for agent communication languages. In: Issues in Agent Communication, Springer-Verlag, pp 31–45

Suguri H, Kodama E, Miyazaki M, Kaji I (2002) Assuring interoperability between heterogeneous multi-agent systems with a gateway agent. In: Proceedings of the 7th IEEE International Symposium on High-Assurance Systems Engineering (HASE 2002), Tokyo, Japan, pp 167–170

Suggested Upper Merged Ontology SUMO (2008) http://www.ontologyportal.org/

Vanderveken D (1990) Meaning and Speech Acts, Principles of Language Use, vol I. Cambridge University Press

Venkatraman M, Singh MP (1999) Verifying compliance with commitment protocols. Autonomous Agents and Multi-Agent Systems 2(3):217–236

Walton DN, Krabbe ECW (1995) Commitment in dialogue: basic concepts of interpersonal reasoning. State University of New York Press

Wooldridge M (2000) Semantic Issues in the Verification of Agent Comunication Languages. Journal of Autonomous Agents and Multi-Agent Systems 3(1):9–31

Yolum P, Singh MP (2004) Reasoning about commitments in the event calculus: An approach for specifying and executing protocols. Annals of Mathematics and Artificial Intelligence 42(1-3):227–253
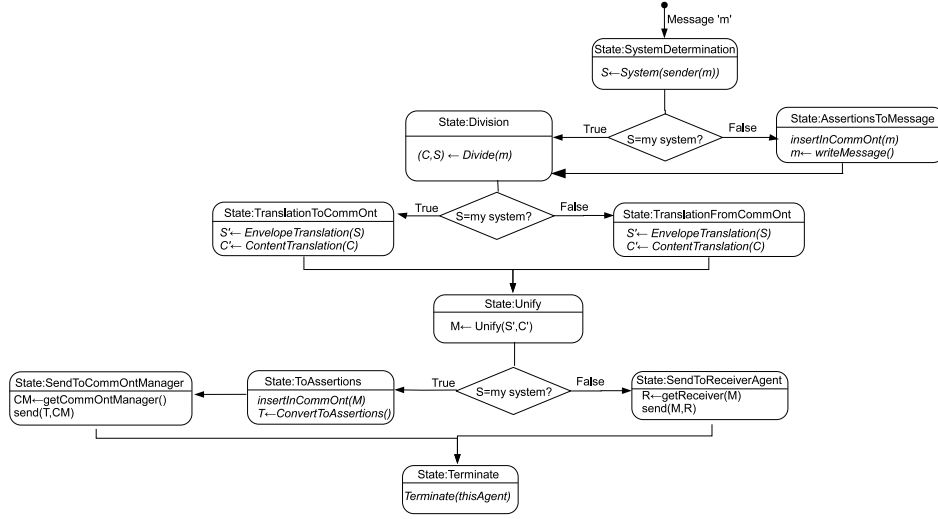
Appendix A.

Figure A.8: State machine of a CommOntManager agent