

A New Mechanism for the Interoperability of Data Systems

Miren I. Bagüés *, Jesús Bermúdez, Arantza Illarramendi, Alberto Tablado,
and Alfredo Goñi **

University of the Basque Country, Donostia, Spain

Abstract. New advances in the areas of Internet and network communications have facilitated in some way the work with distinct data systems located at different places. However, what it is still missing is the possibility of a real and efficient interoperation among those systems. In this paper we present a new mechanism that favors the interoperability of agent based data systems. That mechanism permits an agent of a system 1) to send suitable messages to agents of another system without requiring the establishment of a common communication pattern in advance; and 2) to understand, completely or partially, the messages that are sent by agents of other systems. Moreover, the mechanism also allows the communication among agents that follow different standard languages such as FIPA-ACL and KQML. The main support of the proposed mechanism is an ontology that is divided into three interrelated layers and four different categories.

1 Introduction

The new advances in the areas of Internet and network communications facilitate in some way the work with distinct data systems. However, what it is still missing is the possibility of a real and efficient interoperation among these systems due to their heterogeneity. Besides, nowadays it is increasing the need of the interoperability of these systems in order to interchange information and services.

By data system we mean those systems that must deal with data coming from different sources, that use data management systems and other techniques to work with them and, that provide information to users and other systems. Nowadays there exists a tendency of using agents in data systems because agent technology is broadly recognized as an appropriate technology for approaching problems showing highly distributed nature that need flexible and adaptable

* This work is supported by a grant of the Basque Government.

** All authors are members of the Interoperable DataBases Group, online at <http://siul02.si.ehu.es>. This group is part of the Department of Computer Languages and Systems, at the University of the Basque Country in Spain. This work is mainly supported by the University of the Basque Country, Diputación Foral de Gipuzkoa (cosupported by the European Social Fund) and CICYT [TIN2004-07999-C02-00].

solutions [1]. In this paper we concentrate on these kinds of agent based data systems.

Moreover, currently the communication among agents is, in general, based on the interchange of messages. Agents must be aware in advance of the structure, language and the meaning of the messages in order to deal with them. Although this kind of communication is useful it is also true that it is somehow *limited* because it forces agents to share the same communication pattern. Therefore, the interoperation of agents from different systems is difficult in this scenario. Things get even worse when the agents follow distinct standard communication languages.

In order to allow *communication at a semantic level* among agents that favors the interoperability of data systems we have defined a new mechanism based on the use of an ontology which contains terms related to the communication acts among agents. While messages are the medium through which agents interact they can also be considered as utterances of communication languages. Throughout this paper we use the words “message” and “communication act” as synonyms.

The designed ontology is divided into three interrelated layers and four different categories: *actors* that interact among them using different kinds of messages; *messages* that have different purposes and deal with different kinds of contents; *contents* are the sentences included in the messages; and *subjects* that represent the topic of the messages. Axioms exist in the ontology that describe the interrelationships among these categories.

In the top layer of the ontology, the messages category **Communication Act**, includes terms that describe general communication acts with the aim of being valid in any communication framework. We suggest a conceptualization based on Searle’s classes of *illocutionary acts* [2, 3]: **Assertive**, **Directive**, **Commissive**, **Expressive**, and **Declarative**. Moreover, some specialization of those classes are also appropriate. For example, **Request** and **Inquiry** as different kinds of **Directive**. Of course, the point of this top level conceptualization is a general agreement to serve as a universal reference; therefore, we take it as a work in progress and not as a definitive specification. The **Actor** category represents those entities that send or receive messages. The **Subject** category represents domain specific terms describing the topics of the message. And finally, the **Content** category represents the kind of sentence included in the message.

The middle layer of the ontology is devoted to general purpose communication languages. Specified as subterms of the top layer terms, in the messages category appear terms that two standards, FIPA-ACL [4] and KQML [5], have defined to use as communication acts (e.g. **FIPA-Inform** term for FIPA-ACL and **KQML-Tell** term for KQML are subterms of **Assertive**).

In the bottom layer all the terms described are directly related to a concrete data system that has been considered. Therefore, the first two layers are valid for all data systems. Only the bottom layer must be defined for each data system.

We describe the ontology of communication acts using OWL(Web Ontology Language) [6], a well founded formalism developed by the Web Ontology Work-

ing Group [7] under the auspices of the W3C (World Wide Web Consortium) [8]. In our context, messages are OWL documents. Therefore, messages among agents that use the proposed ontology in this paper have an abstract representation as individuals of a shared universal class of messages. Some researchers have pointed out [9, 10] the benefits of an XML encoding of messages. We believe that an OWL encoding of messages is even more advantageous because it incorporates semantics to the XML syntax. Parsers for OWL documents can be constructed with off-the-shelf tools. The use of OWL technology facilitates software engineering of agents by incorporating the new trends in the Semantic Web [11] technology. Moreover, Semantic Web technologies help in the standardization of the operational semantics of communication acts and offer a well defined infrastructure for sharing ontologies (domain ontologies, device ontologies or whatever).

Furthermore, we claim that the whole communication acts ontology provides interoperability support due to the recognition of communication acts from one language as instances of communication acts in other language. Sometimes the “translation” will not be complete, but partial comprehension of the communication may be useful and preferable to the “not understood” answer given nowadays. Reasoning support provided by the chosen formalism (OWL) may help during the interaction process, as will be explained in section 3.4.

The proposed mechanism, based on the use of the ontology provides the following advantages:

- *Discovering*. An agent that needs to send a message to an agent of another data system can discover the structure of the message that it must send by following a reasoning process with the ontology.
- *Understanding*. An agent can understand, completely or partially, the message sent by an agent of another data system by following a reasoning process with the ontology.
- *Multilanguage*. Two agents that use different communication languages can communicate, although in some situations the communication may be limited. In the current state of the ontology only FIPA-ACL and KQML standards are considered but new ones can be incorporated when necessary.
- *Evolution*. Modifications of the communication terms, at any layer, do not affect agent communications, only the ontology must be fitted.

In the rest of this paper we present first, in section 2, some principles that maintain the proposed mechanism for interoperability. In section 3 we highlight the three layers of our communication acts ontology. Next, in section 4, we show two examples of the use of the interoperability mechanism. Then, in section 5 we comment upon some related works and we finish with the conclusions in section 6.

2 Underlying Principles

The main goal of the present paper is to stress the benefits of achieving an ontological commitment about communication acts performed by cooperating agents.

This is complementary to the development of standards for agent communication languages like KQML or FIPA-ACL. These standards look for general homogeneity through compliance to the standard but are still unable to permit interoperability of agents using different standards, or even the same standard, as is the case with KQML dialects.

Every ontology is defined with a specific purpose. The goal of our ontology is to achieve the interoperability of agent based systems, more particularly interoperability among the agents (of those data systems), that use messages to communicate with each other. We assume agents are committed to cooperate, that is, all agents try to perform the actions that are requested of them. We do not pretend in this paper to present our ontology as the reference ontology (with a definitively established set of terms and relationships among them) for communication among agents. Our purpose is to present it as a proposal because we believe that an accorded ontology for communication is a must. That pragmatic effort should be pointed towards the desired interoperability of agents.

We are conscious of the many possible different approaches to the leading guideline for defining the terms and relationships in the communication acts ontology so, first of all we are going to explain our point of view. It is widely accepted that formalizing the semantics of communication acts avoids ambiguity and establishes a firm ground for implementation purposes. Nevertheless, the multimodal logics¹ used nowadays to express the semantics are often non-computable and it is recognized that the task of validating the implementation based on that logic semantics is very hard. Even, the implementer's intuitive understanding of the communication acts often prevail over the concise semantic definitions [9]. Furthermore, the use of different models and modalities to formalize the semantics of different languages (e.g. KQML and FIPA-ACL) makes them impossible to compare with respect to their formal definition. In this situation, we have adopted a more pragmatic view, we give up some formal properties and modal knowledge. We describe communication acts as somehow more primitive concepts. However, the deep semantics of the communication acts in KQML and FIPA-ACL could be maintained, as in their original specifications, for those agent systems that implement modalities (such as belief, desire, intention).

However, we do not give up on formality, we describe communication acts using OWL that provides suitable computational properties due to its foundations on description logics [16]. Notice that our aim is interoperability, we are not devoted to prove properties of the agent systems, thus we do not need such complex logics mentioned above. The ontology that we propose is a lattice of terms of communication acts and the reasoning system will allow one to compute subsumption between terms and recognition of individuals belonging to

¹ The multimodal logics referred here are a family of first order logics with various modal operators. Particularly modalities for expressing *beliefs*, *desires* and *intentions* of agents performing communication acts. FIPA-ACL uses SL (Semantic language) [4] which is based on [12, 13] and, [14] reports on the modal operators used in KQML. See [15] for more details.

these terms. Finally, the ontology is basically designed for software agents communication but it may also be used for the communication among human and software agents.

3 The Communication Acts Ontology

In this section we present our proposal for the ontology used for communicating among agents, **COMMONT** (hereafter **COMM**unication Acts **ONT**ology) . Firstly, we introduce terms and properties for the top layer. Secondly, we quickly review some of the similarities and differences of KQML and FIPA-ACL with respect to the terms appearing in **COMMONT**. Finally, we focus on the communication acts appearing in the bottom layer of the ontology that are related to a real data system. We explain the design decisions and illustrate the possibilities of the formalism for helping in the management of the interoperation.

3.1 The top layer of the ontology

Through this subsection we show the terms corresponding to the four categories from the ontology in the following order: first those corresponding to **Communication Acts** and then those corresponding to **Content**, **Subject** and **Actor** respectively.

A. Communication Acts

This top layer conceptualization about communication acts should be considered as a framework agreement.

The most general term of this layer is the **CommunicationAct** term. Its individuals are communication acts that may have a sender and a receiver, and depending on its kind they may have a content and a subject. Let us specify it using an abstract syntax in the following manner:

$$\begin{aligned} \text{CommunicationAct} \sqsubseteq & \forall \text{has-sender.Actor} \sqcap \leq 1 \text{ has-sender} \sqcap \\ & \forall \text{has-receiver.Actor} \sqcap \\ & \forall \text{has-content.Content} \sqcap \\ & \forall \text{has-subject.Subject} \end{aligned}$$

For the presentation we prefer this logic notation² instead of the more verbose XML-like notation of OWL. The sentence means that **has-sender**, **has-receiver**, **has-content** and **has-subject** are properties that may be applied to communication acts. Every sender and receiver of an individual in **CommunicationAct** must be an individual of the class **Actor** and there is at most one **has-sender**.

² This notation is common in the description logics field. See [16] for a full explanation. Furthermore, take this statement as a proposal not as a definitive conceptualization

Moreover, every content and subject must be an individual of class **Content** and **Subject** respectively. But it does not mean that necessarily every communication act has a sender, a receiver, a content and a subject (that is a different sentence that should be expressed using a different operator i. e. \exists **has-sender**).

Speech acts theory [2] has been used as a source of inspiration for designing agent communication languages and particularly Searle's classification of *illocutionary acts* [17]. Following this tradition we split up the communication acts into Searle's five classes. **Assertive** is the class of communication acts which commit the sender to the truth of an expressed proposition, **Directive** are those communications which involve getting the receiver to do something, **Commissive** are those which involve committing the sender to some course of action, **Expressive** are those which convey a psychological state of the sender, and **Declarative** are those which bring about the correspondence of the world to the words declared. Some constraints are required for communication acts in those classes. For instance, the content of an assertive must be a proposition, the content of a commissive must be a commitment and the content of a directive must be an action.

$$\begin{aligned} \text{Assertive} &\sqsubseteq \text{CommunicationAct} \sqcap \\ &\quad \forall \text{has-content.Proposition} \\ \text{Directive} &\sqsubseteq \text{CommunicationAct} \sqcap \\ &\quad \forall \text{has-content.Action} \end{aligned}$$

Moreover, it is reasonable to specify particularizations of those classes. Notice that software agents in our context are not prepared to interpret arbitrary communication acts (as is the case in natural language communication), they only recognize individuals on the basis of the values of their properties. Therefore, we want to distinguish a directive asking for information (**Inquiry**), from a directive requesting to perform another kind of action (**Request**); we want to distinguish an assertive informing in response to another message (**Reply-Assertive**), from an assertive that informs autonomously (**Inform**). More specialization should be included if necessary, and notice that disjointness of classes is not assumed unless stated explicitly or logically deduced from statements.

$$\begin{aligned} \text{Request} &\sqsubseteq \text{Directive} \sqcap \\ &\quad \exists \text{has-content.Demand} \\ \text{Inquiry} &\sqsubseteq \text{Directive} \sqcap \\ &\quad \exists \text{has-content.Query} \\ \text{Inform} &\sqsubseteq \text{Assertive} \\ \text{Reply-Assertive} &\sqsubseteq \text{Assertive} \sqcap \\ &\quad \exists \text{in-reply-to.CommunicationAct} \end{aligned}$$

B. Content, Subject and Actor

Communication acts in general may be broken down into the type of action (assert, request, inform, etc.) and the propositional content which specifies details of the action. So the top layer ontology includes a conceptualization of classes of contents. So far, the class **Content** has subclasses **Object**, **Action** and **Proposition** that should be properly conceptualized and subdivided into more specialized classes in the future. For instance, **Query** and **Demand** for the subclass **Action**.

Moreover, every communication act may refer to a topic that we name as its subject. The top layer class **Subject** will be specialized by domain ontologies (e.g. sanitary domain ontology, financial domain ontology,...).

In the context of **COMMONT**, by actors we mean those entities sending or receiving messages. We have divided the category of actors into three subcategories: **SystemWebServices**, **SystemSoftwareAgents** and **SystemHumanAgents**. Those terms respectively describe the general features of the web services defined in the system, the different type of agents that take part in the system and the different human users that are going to interact with the system.

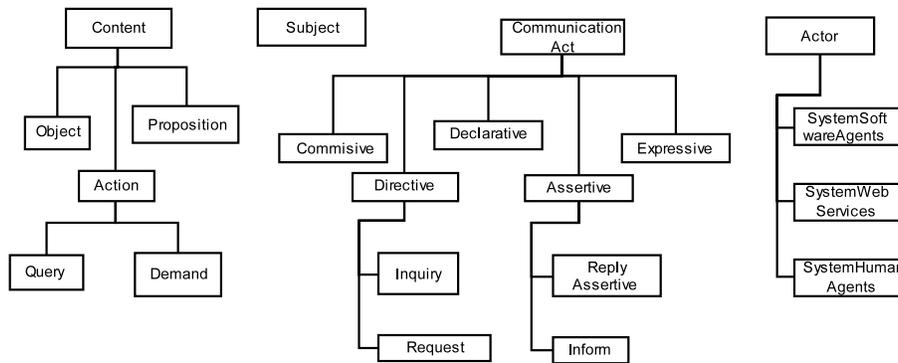


Fig. 1. Top layer of the **COMMONT** ontology.

In figure 1 can be seen the top layer of the ontology described in this subsection.

3.2 The middle layer of the ontology

The top layer ontology can be extended with specific terms that belong to general purpose agent communication languages, like those from **KQML** or **FIPA-ACL**. In the case of **FIPA-ACL**, classes of messages of this language are specified as

subterms of top layer terms according to their semantics. For example,

$$\begin{aligned} \text{FIPA-Inform} &\sqsubseteq \text{Assertive} \\ \text{FIPA-Inform-If} &\sqsubseteq \text{Inform} \sqcap \text{FIPA-Inform} \\ \text{FIPA-Inform-Ref} &\sqsubseteq \text{Inform} \sqcap \text{FIPA-Inform} \\ \text{FIPA-Agree} &\sqsubseteq \text{Reply-Assertive} \sqcap \text{FIPA-Inform} \\ \text{FIPA-Request} &\sqsubseteq \text{Directive} \\ \text{FIPA-Query-If} &\sqsubseteq \text{Inquiry} \sqcap \text{FIPA-Request} \\ \text{FIPA-Query-Ref} &\sqsubseteq \text{Inquiry} \sqcap \text{FIPA-Request} \end{aligned}$$

In the same way, classes of messages of KQML are specified. For example,

$$\begin{aligned} \text{KQML-Tell} &\sqsubseteq \text{Assertive} \\ \text{KQML-Ask-If} &\sqsubseteq \text{Inquiry} \end{aligned}$$

It is of vital relevance for the interoperability aim of the ontology to be able to specify ontological relationships among classes of different standards. For instance, the following axioms represent that `KQML-Ask-If` is equivalent to `FIPA-Query-If` and that `KQML-Achieve` is a subclass of `FIPA-Request`, and `KQML-Tell` is a subclass of `FIPA-Inform`.

$$\begin{aligned} \text{KQML-Ask-If} &\equiv \text{FIPA-Query-If} \\ \text{KQML-Achieve} &\sqsubseteq \text{FIPA-Request} \\ \text{KQML-Tell} &\sqsubseteq \text{FIPA-Inform} \end{aligned}$$

Nevertheless, KQML and FIPA-ACL differ substantially in their semantic framework [18] to the point that, in general, a complete and accurate translation between them is not possible. But it is also true that they share basic concepts and principles to such an extent that we can define ontological relationships in the context of the interoperability of data systems.

Conceptually KQML and FIPA-ACL consider two components in a message: the message class component and the message content component. Both also claim to be independent of the content language and promote the message class component as being responsible for determining the kinds of communication acts they consider. Nevertheless, a more in-depth study of both languages reveals some disagreement with respect to the limits of what to include in the message class component or in the message content component. In fact, the boundaries of the two components are not clearly cut. For example, when a FIPA agent wants to tell another agent to achieve one goal G , the agent will

send a message of class `FIPA-Request`, with a content expression referring to an action `achieve` (from an ontology of actions) and with the goal G as a parameter. Instead, a KQML agent trying to communicate the same thing will send a message of class `KQML-Achieve` with the goal G in the content expression. Luckily, that relationship can be expressed in the `COMMONT` ontology using the following sentence:

$$\text{KQML-Achieve} \equiv \text{FIPA-Request} \sqcap \\ \exists \text{has-content.}\{\text{achieve}\}$$

We think that the decision of what to include in the message class component and what to include in the content component is biased by the purpose of the language. General purpose languages tend to design more general classes of messages shifting to the content component the responsibility of expressing more concrete things. In KQML design, the repertoire of classes of messages is consciously not closed in order to cope freely with this situation. But experience has shown that various KQML dialects have emerged, and unfortunately they are not interoperable with the genuine KQML. This lack of interoperability is another reason that guarantees the interest of the `COMMONT` ontology we are proposing. On the other side, typical agent based data systems usually deal with a limited collection of kinds of messages. Shifting the responsibility of interpreting those messages to the content component may not be the most appropriate decision. Taking that in mind we have designed the bottom layer of the ontology.

3.3 The bottom layer of the ontology

It is often the case that every single agent based system uses a limited collection of communication acts that constitute its particular communication language. Some of those communication acts can be defined as particularizations of existing standards in the middle layer and maybe some others as particularizations of top layer terms. Nevertheless their specification in our communication ontology will favor the interoperability with related agent based systems.

We are going to present the terms for this layer using a concrete data system, `AINGERU`³: an agent based data system for a new way of tele-assistance for elderly people. The `AINGERU` system, apart from supporting the functionalities provided by current tele-assistance services, also offers: an active assistance by using agents that behave in the face of anomalous situations without a direct intervention of the user; an anywhere and anytime assistance by using wireless communications and PDAs (Personal Digital Assistant); and the monitoring of personal vital signs by using sensors that capture the values of those signs and feed a decision support system that analyzes them and generates an alarm when necessary.

³ `AINGERU` is the word in the Basque language for expressing the notion of a guardian angel.

After completing the requirements analysis of the system, three major classes of messages were identified (among others that we do not explain here so as not to include too much detail): **A-Request**⁴, **A-QueryRef** and **A-InformResult**.

- **A-Request** includes the messages demanding the receiver to perform an action
- **A-QueryRef** includes the messages asking the receiver for some information
- **A-InformResult** includes the messages sending results in reply to some request.

More specifically, the value of the property **has-content** for every message in the class **A-Request** is an action from the class **Demmand** named **do**. The value of **has-content** for a message in **A-QueryRef** is an action from the class **Query** named **give-me**; and, respectively, the value of **has-content** for a message in **A-InformResult** is a content named **collection**.

```

A-Request ⊆ Request ⊓
    ∃has-content.{do} ⊓ ≤ 1 has-content ⊓
    ∃has-subject.Subject ⊓ ≤ 1 has-subject
A-QueryRef ⊆ Inquiry ⊓
    ∃has-content.{give-me} ⊓ ≤ 1 has-content ⊓
    ∃has-subject.Subject ⊓ ≤ 1 has-subject
A-InformResult ⊆ Reply-Assertive ⊓
    ∃has-content.{collection} ⊓ ≤ 1 has-content ⊓
    ∃has-subject.Subject ⊓ ≤ 1 has-subject

```

Notice that now the interpretation of a message from **A-Request** (respectively from the other two classes) depends entirely on its subject.

Moreover, considering two components in a message (as FIPA-ACL and KQML would consider them) we have advocated for including in the message class component the explicit representation of the specific type of communication act, depending on the kind of the subject of the message. Therefore, each of the previous major classes are subdivided into different subclasses (see figure 2 for a fragment of the ontology and [19] for a broader explanation). For instance,

```

MedicineModify ⊆ A-Request ⊓
    ∃has-subject.Medicine
LocationQuery ⊆ A-QueryRef ⊓
    ∃has-subject.Location

```

⁴ **A-Request** means **Aingeru-Request** and analogously for all the **A-** prefixes.

VitalSignInform \sqsubseteq A-InformResult \sqcap
 \exists has-subject.VitalSign

A **MedicineModify** message is used to request a change in the medicines prescription, a **LocationQuery** message asks for the coordinates of the physical location of a user, and a **VitalSignInform** message informs about the values of the vital signs of a person. This type of representation facilitates the interpretation of messages by the agents. In the subsection 3.4 we explain in more detail the advantages of this approach.

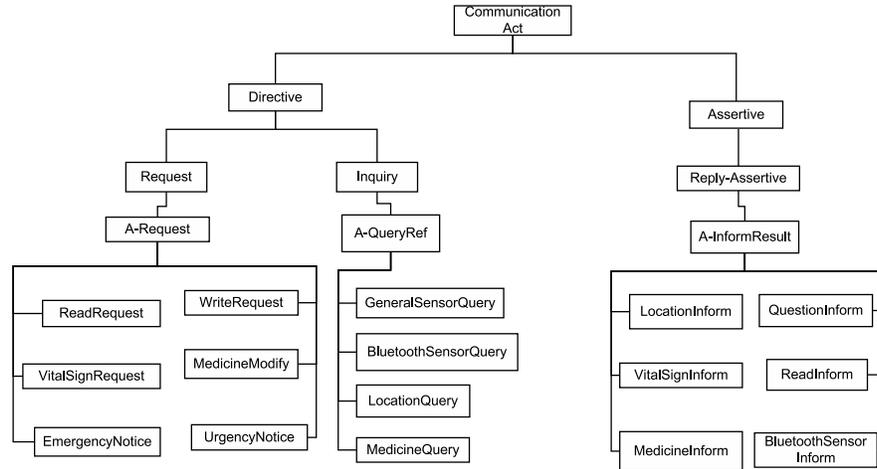


Fig. 2. Fragment of the message category of COMMONT.

In this bottom layer of the ontology also appear terms related with the subject category and actor category. Concerning subject category, in figure 3 can be observed an example of a fragment of the ontology of subjects in AINGERU. Agent based data systems developed on different domains will use different ontologies of subjects, but those developed on the same domain should share some concepts in the ontology of subjects if they want to interoperate.

With respect to the category of actors in our application scenario (see figure 3), **HumanAgents** includes subclasses for users of the AINGERU system as well as for those people concerned with the user assistance, from sanitary people to relatives. **SoftwareAgents** includes specialized classes for agents described taking into account their location and goals (for example, whether they work in a PDA or in a computer, are they attending a sensor or interacting with an ontology, and so on). **AingeruWebServices** is a subclass of **SystemWebServices** that describes the web services exported by AINGERU. For example, the web service **WebLocation** permits one to obtain the location of the user of the PDA.

Every service in `AingeruWebServices` has a property `provideService` whose value is an DAML-S description of the Web Service that these agents provide.

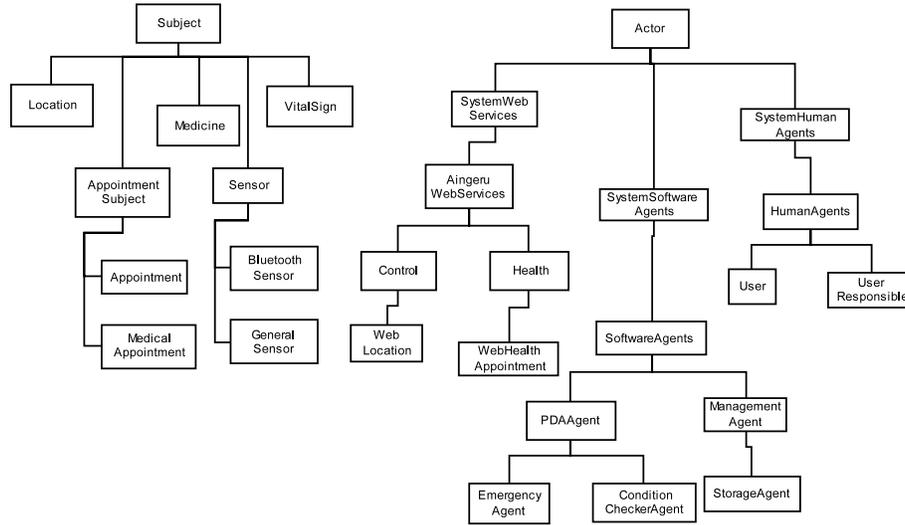


Fig. 3. Fragment of the Subject and Actor categories of COMMOnt.

Having web services as a part of our ontology permit us to describe them at a semantic level independently of the language in which they are expressed. Due to this it is easier to find the adequate service in each case. If an external agent wants to use one of the services that the system exports as web services, it has two different ways of doing it: describing semantically what it wants to do (the reasoning mechanism of the ontology will infer which is the web service it has to use and which are the attributes it needs); or, using the standards for Web Services.

3.4 Benefits of the explicit representation of messages using OWL

Now we want to show some more benefits of representing explicitly the messages as individuals of OWL classes. The description of classes can include necessary constraints for the individuals in a class as well as sufficient conditions for individuals to be recognized as members of a particular class. All the logic sentences stated so far (except those with the equivalence \equiv symbol) express necessary constraints for the individuals in the class named to the left of the \sqsubseteq symbol. Moreover, it is possible to take advantage of the formalism to state axioms that specify minimal sufficient conditions to recognize that an individual belongs to a certain class. For instance:

$$\text{Inquiry} \sqcap \exists \text{has-subject.Location} \sqsubseteq \text{LocationQuery}$$

Reply-Assertive $\sqcap \exists$ has-subject.Medicine \sqsubseteq MedicineInform

Using this capability of expressiveness of the OWL formalism and the supported reasoning capability, it is possible to discover the most specific class of a message and the collection of properties that are applicable to it. That information can be used by an external agent for the construction of a message that must be sent to an agent of a different system. Basically, the external agent only needs to know the top layer class of the message it wants to send in addition to the subject that it is about. Examples that illustrate this task are presented in section 4. Notice that this process is done at runtime, allowing an agent to interoperate with a different kind of agent without being previously aware of how to do it. Sometimes the message built may be partially understood. But this gives the agents the opportunity to react, which is hopefully better than receiving a “non understood message” response.

4 Examples of the use of the CommOnt Ontology

We show in this section two different scenarios in which the agents of different systems interoperate between them. The main goal of the section is to show through two examples how our proposal helps in the task of the interoperation among agents.

4.1 Interoperation of an external agent with Aingeru

The external agent belongs to an information system of a hospital and its task is to know where the user is because an alarm has been received from the **EmergencyAgent** in the PDA of the user.

There are two main steps that the external agent must do to accomplish its goal: i) to discover how to communicate with AINGERU agents and ii) to contact the adequate AINGERU agent. In the first step the ontology plays an important role. The external agent knows *what it wants* to communicate but it does not know *how* to interchange messages with AINGERU agents. Therefore, it uses the COMMONT ontology to obtain the properties that an AINGERU message must have. In this step, we can observe how our mechanism favors the interoperability with external agents. The external agent asserts the following statements to create a new message⁵ **m1** that asks for the location of a user:

```
m1 <type> Inquiry
m1 <has-subject> loc1
loc1 <type> Location
```

Then the reasoning system will infer that **m1** is an instance of the **LocationQuery** class.

⁵ We use an abstract syntax.

m1 <type> LocationQuery

Next the external agent asks the ontology for the attributes associated to LocationQuery. It will receive the following list of attributes: `ident`, `has-sender`, `has-receiver`, `has-subject`, `has-content`.

In the second step, the external agent needs to discover which and where the specialist AINGERU agent is. That agent is to whom the external agent must send the message obtained at step 1. There are several agencies in AINGERU that offer facilities to discover which agent offers a certain service. They are called Service Discovery facilities and are based on the DAML-S [20] Web Services Description Language. In our example, the external agent asks (to the Service Discovery) for the particular agent that is in charge of knowing the location of the user.

In this step we can see the flexibility that AINGERU offers to deal with different agents. With the information obtained in both steps the external agent can create the LocationQuery message that must be sent (see figure 4).

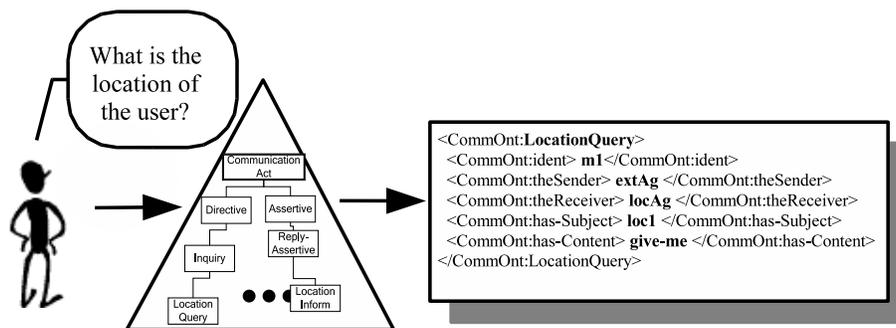


Fig. 4. Using the ontology to build the message.

Once the message is created and the external agent knows to which agent needs to send its request, it establishes the communication and waits for the response. When the external agent receives a reply message it asserts the statements within that message in order to understand it. Then the reasoning system will infer that the message is an instance of the LocationInform class.

m2 <type> LocationInform

Next the external agent asks the ontology for the values of the properties associated to m2. It will receive the following pairs of attribute-value:

```

ident:m2
has-sender:locAg
has-receiver:extAg
has-content:collection
has-subject:loc1
loc1.theLocation:43°18'26", -2°0'41"

```

With this information the external agent can understand the message that it has received, which is: The user location is 43°18'26", -2°0'41" (see figure 5).

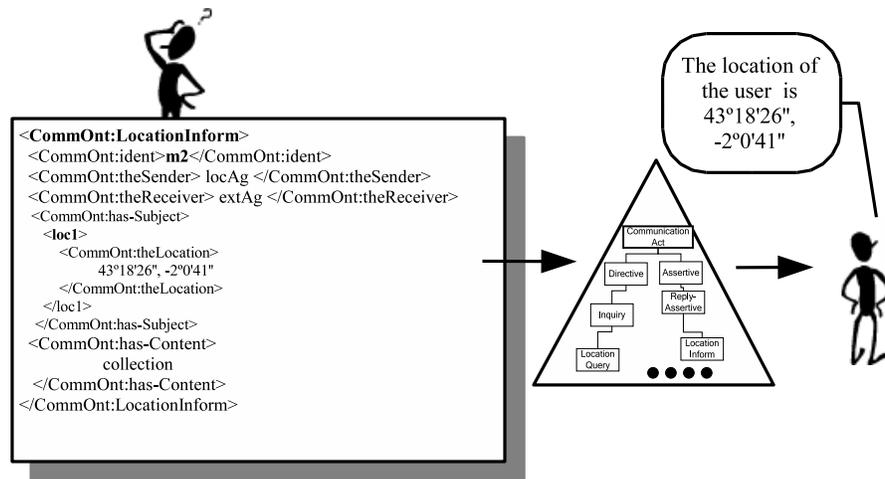


Fig. 5. Using the ontology to understand the result.

4.2 Interoperation between an agent that uses FIPA-ACL messages and another that uses KQML messages

In this example we want to show how interoperation using messages from different languages can be achieved through our COMMONT ontology. Let us suppose an agent F, that uses FIPA-ACL messages, requests to an agent K, that uses KQML messages, to achieve a temperature of 21 degrees in a room. F will send the following FIPA-ACL message mF1:

```

(request
  :sender (agent-identifier : name F)
  :receiver (set (agent-identifier : name K))
  :content ‘‘achieve = (temperatureRoom,21)’’)

```

The abstract representation of that message using COMMONT involves the following statements:

```

mF1 <type> FIPA-Request
mF1 <has-sender> F
mF1 <has-receiver> K
mF1 <has-content> achieve
mF1 <has-subject> (temperatureRoom,21)

```

Then, due to the axiom $KQML\text{-Achieve} \equiv FIPA\text{-Request} \sqcap$
 $\exists \text{has.content.}\{achieve\}$, mF1 is recognized as a message in $KQML\text{-Achieve}$,
and the agent K understands it completely and is able to process it.

Let us suppose now that agent F sends to agent K the following FIPA-ACL
message mF2:

```

(request
  : sender (agent-identifier : name F)
  : receiver (set (agent-identifier : name K))
  : content
    ‘‘((action (agent-identifier : name K)
      (inform-if
        : sender (agent-identifier : name K)
        : receiver (set (agent-identifier : name F))
        : content ‘‘empty mailbox’’)))’’’)

```

Then the abstract representation includes, among others, the following state-
ments:

```

mF2 <type> FIPA-Request
mF2 <has-content> action

```

In this case there is not enough information for recognizing mF2 as a message
in any KQML class. The most likely continuation of this conversation in nowa-
days agent systems would be a “not understood” message and the end of the
conversation. But due to the axiom $FIPA\text{-Request} \sqsubseteq Directive$ in $COMMONT$
ontology, mF2 <type> $Directive$ is deduced.

Since every agent using $COMMONT$ know the terms in the top layer of the
ontology, agent K understands mF2 only partially (K understands that mF2 is
a $Directive$, but no more). Trying to cooperate, agent K sends to agent F a
 $KQML\text{-Tell}$ message informing about its own capabilities on performing direc-
tives. Due to $KQML\text{-Tell} \sqsubseteq FIPA\text{-Inform}$, agent F is able to completely under-
stand the reply from K, and to discover that `empty` is a predicate about which
agent K can be asked. Therefore, agent F decides to deliver to agent K a message
from another class, like the following mF3:

```

(query-if
  : sender (agent-identifier : name F)
  : receiver (set (agent-identifier : name K))
  : content ‘‘empty mailbox’’)

```

Now the abstract representation includes the following statements:

```
mF3 <type> FIPA-Query-If
mF3 <has-content> empty
mF3 <has-subject> mailbox
```

Since the axiom $\text{KQML-Ask-If} \equiv \text{FIPA-Query-If}$ is in `COMMONT`, `mF3 <type>` `KQML-Ask-if` is deduced and, therefore, agent K is able to understand `mF3` completely and to process it satisfactorily.

5 Related works

Our work is complementary to the development of standards for agent communication languages like KQML or FIPA-ACL (well summarized in [9]). These standards look for general homogeneity through compliance to the standard. In the following we mention some other related works. In [21] a formal framework is presented for agents to negotiate the semantics of their ACL at runtime. A *semantic space* provides a means to systematically analyze inter-agent communication. A point in this space can be identified with a particular communication act. They advocate for the specification of message semantics in a common formalism. We view our `COMMONT` ontology as a complementary effort. Any interesting relationship between classes of messages discovered within their analysis method can be explicitly encoded in `COMMONT`. Therefore, our proposal framework acts as a compiled representation of messages classification.

The following two papers [22, 23] present a semantic communication stack that includes the message class component and the content component among others. They emphasize in the need for agreements on the formalisms used for each component and in the need to manage the dependencies between components. They also point out that one of the major challenges is to find out the right trade-off between implicit versus explicit semantics describing abstractions in each component. In our case the content component and the message component can be described using the same formalism. Moreover, in [24] they propose an abstract ontology representation (AOR) for capturing abstract models of communication related knowledge (domain models, agent communication languages, content languages and models of how these interact). Our `COMMONT` ontology can be considered as part of that AOR.

A different approach for interoperability, based on the inclusion of preformatted message templates within the advertised capability description of agents is presented in [25]. The shallow-parsing template approach presented in that paper relaxes the constraint that agents share a common language for describing the content and format of messages. One main difference of that approach with respect to ours is their major emphasis on syntactic aspects.

Furthermore, experience reported in [26] suggests that there is a strong overlap in the communication acts required by many agent systems and therefore they claim that a small comprehensive set would be sufficient for many multi-agent systems. We agree with them and furthermore, the `COMMONT` ontology permits the description of different communication acts whilst maintaining the relevant relationships. Finally, in [26] it is noticed that the question of how much

content information should be pushed to the message component is an important research issue. Our work goes in that direction.

6 Conclusions

The heterogeneous data systems interoperability is a need that is manifested each time with a greater intensity in different fields. However, currently the interoperation is very restricted and there is a long way to traverse until a real and efficient interoperation is possible.

In this paper we have presented one mechanism that takes us one step closer to achieving that interoperability by allowing the communication among agents of different data systems.

The mechanism is based on one ontology that describes the communication acts among agents of different data systems. That ontology is made up of three layers that try to collect communication acts at different levels of abstraction. The terms of the top layer should be defined by experts in the agents communication area and they should be considered as reference terms for communication. The elements of the middle layer reflect the terms that different standard communication languages have defined.

Finally, in the bottom layer, we advocate for defining terms that describe messages used in each concrete data system. This type of message descriptions facilitates their interpretation because it is possible to reason with them.

References

1. Jennings, N.R.: An agent-based approach for building complex software systems. *Commun. ACM* **44** (2001) 35–41
2. Searle, J.R.: *Speech acts*. Cambridge University Press (1969) New York.
3. Traum, D.R.: *Speech acts for dialogue agents*. In Wooldridge, M., Rao, A., eds.: *Foundations of Rational Agency*. Kluwer Academic Publishers (1999) 169–201
4. Foundation For Intelligent Physical Agents: *FIPA Communicative Act Library Specification*. (2002) <http://www.fipa.org/specs/fipa00037/SC00037J.html>.
5. Finin, T., Labrou, Y., Mayfield, J.: *KQML as an agent communication language*. In Bradshaw, J., ed.: *Software Agents*. MIT Press (1997)
6. World Wide Web Consortium: *OWL Web Ontology Language Reference*. (2003) <http://www.w3.org/TR/owl-ref>.
7. Web-Ontology (WebOnt) Working Group: (2004) <http://www.w3.org/2001/sw/WebOnt/>.
8. World Wide Web Consortium: (2004) <http://www.w3.org>.
9. Labrou, Y.: *Standardizing agent communication*. In Marik, V., Stepankova, O., eds.: *Multi-Agent Systems & Applications. Advanced Course on Artificial Intelligence (ACAI-01)*, Springer-Verlag (2001) 74–97
10. Grosz, B., Labrou, Y.: *An approach to using xml and a rule-based content language with an agent communication language*. In Dignum, F., Greaves, M., eds.: *Issues in Agent Communication. Lecture Notes in Artificial Intelligence*. Springer-Verlag (2000) 96–117

11. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* **284(5)** (2001) 34–43
12. Cohen, P., Levesque, H.: Intentions is choice with commitment. *Artificial Intelligence* **42** (1990) 213–231
13. Bretier, P., Sadek, M.D.: A rational agent as the kernel of a cooperative spoken dialogue system: implementing a logical theory of interaction. In Müller, J., Wooldridge, M., Jennings, N., eds.: *Intelligent Agents III*. Volume 1193. Springer-Verlag, Berlin, Germany (1997) 189–204
14. Labrou, Y., Finin, T.: Semantics and conversations for an agent communication language. In: *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, Nagoya, Japan (1997) 584–591
15. Wooldridge, M.: *Reasoning about Rational Agents*. MIT Press (2000)
16. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: *The Description Logic Handbook. Theory, Implementation and Applications*. Cambridge University Press (2003)
17. Searle, J.R.: A classification of illocutionary acts. *Language in Society* **5** (1976) 1–23
18. Labrou, Y., Finin, T., Peng, Y.: Agent communication languages: the current landscape. *IEEE Intelligent Systems* **14** (1999) 45–52
19. Bagüés, M.I., Bermúdez, J., Illarramendi, A., Tablado, A., Goñi, A.: Using ontologies in the development of an innovating system for elderly people tele-assistance. In: *Proceedings of the International Conference on Ontologies, Databases and Applications of SEMantics*. Volume 2888 of *Lecture Notes on Computer Science*., Springer-Verlag (2003) 889–905 Sicily, Italy.
20. DAML Services: (2004) <http://www.daml.org/services>.
21. Reed, C., Norman, T.J., Jennings, N.R.: Negotiating the semantics of agent communication languages. *Computational Intelligence* **18** (2002) 229–252
22. Willmott, S., Dale, J., Charlton, P.: *Agent Communication Semantics for Open Environments*. Foundation for Intelligent Physical Agents. (2002) Input document f-in-00066. <http://www.fipa.org/docs/input/f-in-00066/>.
23. Calisti, M.: Abstracting communication in distributed agent-based systems. In: *Proceedings of the 16th European Conference on Object-Oriented Programming*, Malaga, Spain (2002)
24. Willmott, S., Constantinescu, I., Calisti, M.: Multilingual agents: Ontologies, languages and abstractions. In: *Proceedings of First International Workshop on Ontologies in Agent Systems*, Montreal, Canada (2001)
25. Payne, T., Singh, R., Sycara, K.: Communicating agents in open multi-agent systems. In: *First GSFC/JPL Workshop on Radical Agent Concepts (WRAC)*, McLean, VA, USA (2002)
26. Nodine, M.H., Unruh, A.: Facilitating open communication in agent systems: The InfoSleuth infrastructure. In: *Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages*. (1997) 281–295