Ontological Commitments on the Classes of Messages: an Step towards Agents Semantic Interoperability

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

University of the Basque Country, Donostia, Spain

Abstract. Nowadays it is increasing the tendency of using agents on behalf of information systems in their interactions with other autonomous information systems. Moreover, there exists an actual interest on getting these interactions at a semantic level, that is, beyond the syntactic level interaction provided by the XML format standard.
In this paper we present a mechanism that gives an step forward in the global goal of achieving a semantic interoperability among agents. The mechanism is based on ontological commitments on the classes of messages interchanged by agents of different information systems in an scenario where agents act in a sincere, helpful and liberal way.
Furthermore those messages are considered as individuals of OWL classes, and we take advantage of the reasoning supporting OWL ontologies.

1 Introduction

The new advances in the areas of Internet and network communications allow closer relationship among distinct information 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 there exists a tendency of using agents in the information systems because agent technology is broadly recognized as an appropriate technology for approaching problems showing highly distributed nature that need flexible and adaptable solutions [1].

Currently the communication among agents is, in general, based on the interchange of messages. Different information systems have incorporated different classes of messages as their *Agent Communication Languages* (ACL). Therefore, the interoperation of agents from different systems is difficult.

* 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 and Diputación Foral de Gipuzkoa (cosupported by the European Social Fund) and CICYT [TIN2004-07999-C02-00].

In order to allow *communication at a semantic level* among agents (let us also call them information agents), that favors the interoperability of information systems, we have defined a new mechanism based on the use of an ontology which contains terms related to the communication acts among information agents (throughout this paper we use the words "message" and "communication act" as synonyms). The main goal of the present paper is to stress the benefits of achieving an ontological commitment about communication acts performed by cooperating information agents.

In the considered context, we make the following assumptions about the behavior of information agents. We assume that information agents are *sincere*: they communicate their knowledge with veracity; *helpful*: they always try to perform what they are asked for, if it is within their capabilities; and *liberal*: they have no objection to communicate with agents of different models, they try to accommodate to the prerequisites. So far, we consider that information agents are not mentalistic agents (i.e. their behavior is not based on explicit representation of beliefs, desires, intentions).

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*; and *subjects* that represent the topic of the messages. Axioms exist in the ontology that describe the interrelationships among these categories.

We do not pretend in this paper to present our ontology as a definitive ontology 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. Efforts made in the development of upper level ontologies such as DOLCE [2] and SUMO [3] are to be considered. That pragmatic effort should be pointed towards the desired interoperability of agents.

The core of the ontology that we propose is a lattice of terms for communication acts. We describe the terms using OWL(Web Ontology Language) [4]. In our context, properties of messages are represented with OWL statements. 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. OWL provides suitable computational properties due to its foundations on description logics [5]. The reasoning system allows to compute subsumption between terms and recognition of individuals belonging to these terms. Some researchers have pointed out [6, 7] 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. The use of OWL technology facilitates software engineering of agents by incorporating the new trends in the Semantic Web [8] 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 messages from one language as instances of messages 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.

As we show in the rest of the paper 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 information 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 information 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.
- *Evolution.* Insertions and modifications of 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, a highlight of the three layers of our communication acts ontology. Next, in section 3, we show two examples of the use of the interoperability mechanism. Then, in section 4 we comment upon some related works and we finish with the conclusions in section 5.

2 The Communication Acts Ontology

In this section we present our proposal for the COMMunication Acts ONTology hereafter CommOnt. Firstly, we introduce terms and properties for the top level. Secondly, we quickly review some of the similarities and differences of two standard agent communication languages, KQML [9] and FIPA-ACL [10] with respect to the terms appearing in CommOnt. Finally, we focus on the communication acts appearing in the bottom level of the ontology that are related to a concrete information system. We explain the design decisions and illustrate the possibilities of the formalism for helping in the management of the interoperation.

2.1 The top level of the ontology

Through this subsection we show the terms corresponding to the four categories represented in the ontology in the following order: first those corresponding to `CommunicationAct` and then those corresponding to `Content`, `Subject` and `Actor` respectively.

A. Communication Acts

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

The most general term of this level 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:

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

For the presentation we prefer this logic notation[1] 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`. 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 [11] has been used as a source of inspiration for designing agent communication languages and particularly Searle's classification of *illocutionary acts* [12]. 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 and the content of a directive must be an action.

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

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

[1] This notation is common in the description logics field. See [5] for a full explanation. Furthermore, take this statement as a proposal not as a definitive conceptualization.

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 the statements.

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

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 level 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 level 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.

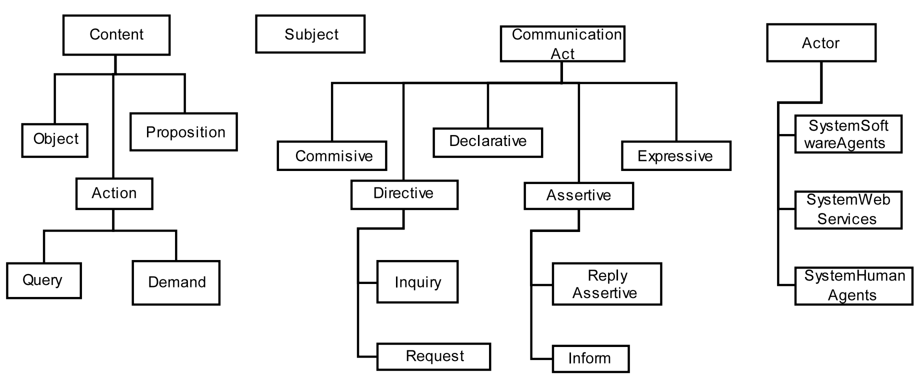

Fig. 1. Top level of the CommOnt ontology.

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

2.2 The middle level of the ontology

The top level 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 level terms according to their semantics. For example,

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

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

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

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, `KQML-Ask-If` is equivalent to `FIPA-Query-If` or `KQML-Achieve` is a subclass of `FIPA-Request`.

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

Nevertheless, KQML and FIPA-ACL differ substantially in their semantic framework [13] 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 information agents. A careful specification of their semantics within a common formalism, as suggested in [14], would show eventual relationships that could be explicitly encoded into CommOnt.

Furthermore one specific aspect from the standard languages have guided us towards the enhancing of the message class layer. Conceptually KQML and FIPA-ACL consider two layers in a message: the message class layer and the message content layer. Both also claim to be independent of the content language and promote the message class layer 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 layer or in the message content layer. In fact, the boundaries of the two layers 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:

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

We think that the decision of what to include in the message class layer and what to include in the content layer is biassed by the purpose of the language. General purpose languages tend to design more general classes of messages shifting to the content layer the responsibility of expressing more concrete things. In KQML design, the repertoire of classes of messages is consciously left open 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 information systems usually deal with a limited collection of kinds of messages. Shifting the responsibility of interpreting those messages to the content layer may not be the most appropriate decision. Taking that in mind we have designed the bottom level of the ontology.

2.3 The bottom level 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 level and maybe some others as particularizations of top level terms. Nevertheless their specification in our communication ontology will favor the interoperability with related agent based information systems.

We are going to present the terms for this level using a concrete information system, Aingeru[2]: an agent based information 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.

[2] 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`[3], `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 `Demand` 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`.

$$
\begin{aligned}
\texttt{A-Request} \sqsubseteq\ & \texttt{Request} \sqcap \\
& \exists\, \texttt{has-content}.\{\texttt{do}\} \sqcap \leq 1\ \texttt{has-content} \sqcap \\
& \exists\, \texttt{has-subject}.\texttt{Subject} \sqcap \leq 1\ \texttt{has-subject} \\
\texttt{A-QueryRef} \sqsubseteq\ & \texttt{Inquiry} \sqcap \\
& \exists\, \texttt{has-content}.\{\texttt{give-me}\} \sqcap \leq 1\ \texttt{has-content} \sqcap \\
& \exists\, \texttt{has-subject}.\texttt{Subject} \sqcap \leq 1\ \texttt{has-subject} \\
\texttt{A-InformResult} \sqsubseteq\ & \texttt{Reply-Assertive} \sqcap \\
& \exists\, \texttt{has-content}.\{\texttt{collection}\} \sqcap \leq 1\ \texttt{has-content} \sqcap \\
& \exists\, \texttt{has-subject}.\texttt{Subject} \sqcap \leq 1\ \texttt{has-subject}
\end{aligned}
$$

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

Moreover, considering two layers in a message (as FIPA-ACL and KQML consider them) we have advocated for including in the message class layer 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 [15] for a broader explanation). For instance,

$$
\begin{aligned}
\texttt{MedicineModify} &\sqsubseteq \texttt{A-Request} \sqcap \exists\, \texttt{has-subject}.\texttt{Medicine} \\
\texttt{LocationQuery} &\sqsubseteq \texttt{A-QueryRef} \sqcap \exists\, \texttt{has-subject}.\texttt{Location} \\
\texttt{VitalSignInform} &\sqsubseteq \texttt{A-InformResult} \sqcap \exists\, \texttt{has-subject}.\texttt{VitalSign}
\end{aligned}
$$

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 2.4 we explain in more detail some advantages of this approach.

[3] `A-Request` means `Aingeru-Request` and analogously for all the `A-` prefixes.

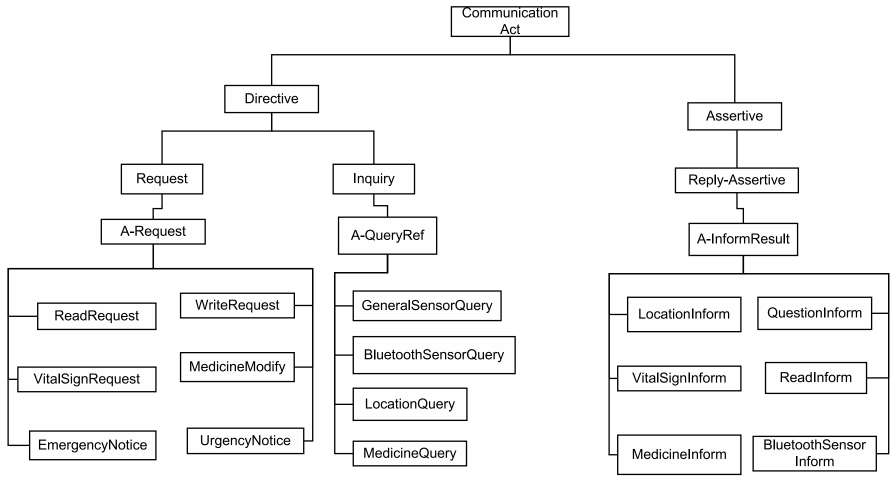

Fig. 2. Fragment of the message category of CommOnt.

In this bottom layer of the ontology also appears terms related with the subject category and actors category. Concerning subject category, agent based information 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, `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 a DAML-S description of the Web Service provided.

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.

2.4 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:

$$\texttt{Inquiry} \sqcap \exists\, \texttt{has-subject.Location} \sqsubseteq \texttt{LocationQuery}$$
$$\texttt{Reply-Assertive} \sqcap \exists\, \texttt{has-subject.Medicine} \sqsubseteq \texttt{MedicineInform}$$

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 level 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 3. 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.

3 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.

3.1 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 message[4] `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.

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 [16] 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.

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 `m2` 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.

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`,($43^o18'26''$, $-2^o0'41''$)}.

With this information the external agent can understand the message that it has received which is: the user location is $43^o18'26''$, $-2^o0'41''$.

3.2 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`:

---

[4] We use an abstract syntax.

```
(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-content>achieve. Then, due to the axiom
KQML-Achieve $\equiv$ FIPA-Request $\sqcap$ $\exists$ has.content.$\{$achieve$\}$, mF1 is recognized as a message in KQML-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 statements: 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 nowadays agent systems would be a "not understood" message and the end of the conversation. But due to the axiom FIPA-Request $\sqsubseteq$ Directive in CommOnt ontology, mF2<type>Directive is deduced.

Since every agent using CommOnt knows the terms in the top level 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-Tell message informing about its own capabilities on performing directives. Due to KQML-Tell $\sqsubseteq$ FIPA-Inform, agent F is able to completely understand 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 statement: mF3 <type> FIPA-Query-If. Since the axiom KQML-Ask-If $\equiv$ FIPA-Query-If is in Comm-

Ont, `mF3 <type> KQML-Ask-if` is deduced and, therefore, agent `K` is able to understand `mF3` completely and to process it satisfactorily.

4 Related works

Our work is complementary to the development of standards for agent communication languages like KQML or FIPA-ACL (well summarized in [6]). These standards look for general homogeneity through compliance to the standard. In the following we mention some other related works. In [14] 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 [17, 18] present a semantic communication stack that includes the message layer and the content layer among others. They emphasize in the need for agreements on the formalisms used in each layer and in the need to manage the dependencies between layers. 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 layer. In our case the content layer and the message layer can be described using the same formalism. Moreover, in [19] 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 [20]. 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 [21] 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 multiagent systems. We agree with them and furthermore, the CommOnt ontology permits the description of different communication acts whilst maintaining the relevant relationships. Finally, in [21] it is noticed that the question of how much content information should be pushed to the message layer is an important research issue. Our work goes in that direction.

5 Conclusions

Nowadays it is widely recognized the necessity of defining mechanisms that permit agent to interact among them, even they do not share a common understanding of the world. In this paper we have presented a mechanism that gives an step forward in the global goal of achieving a semantic interoperability among agents.

The mechanism is based on one ontology that describes the communication acts among agents of different information 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 level 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 level reflect the terms that different standard communication languages have defined.

Finally, in the bottom level, we advocate for defining terms that describe messages used in each concrete information 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. Communication of the ACM **44** (2001)
2. DOLCE: a Descriptive Ontology for Linguistic and Cognitive Engineering (2004) `http://www.loa-cnr.it/DOLCE.html`.
3. SUMO Ontology: (2004) `http://ontology.teknowledge.com/`.
4. World Wide Web Consortium: OWL Web Ontology Language Reference. (2003) `http://www.w3.org/TR/owl-ref`.
5. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook. Theory, Implementation and Applications. Cambridge University Press (2003)
6. 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)
7. Grosof, 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
8. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **5** (2001)
9. Finin, T., Labrou, Y., Mayfield, J.: KQML as an agent communication language. In Bradshaw, J., ed.: Software Agents. MIT Press (1997)
10. Foundation For Intelligent Physical Agents: FIPA Communicative Act Library Specification. (2002) `http://www.fipa.org/specs/fipa00037/SC00037J.html`.
11. Searle, J.R.: Speech acts. Cambridge University Press (1969) New York.
12. Searle, J.R.: A classification of illocutionary acts. Language in Society **5** (1976) 1–23

13. Labrou, Y., Finin, T., Peng, Y.: Agent communication languages: the current landscape. IEEE Intelligent Systems (1999) 45–52
14. Reed, C., Norman, T.J., Jennings, N.R.: Negotiating the semantics of agent communication languages. Computational Intelligence **18** (2002)
15. 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.
16. DAML Services: (2004) `http://www.daml.org/services`.
17. 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/`.
18. Calisti, M.: Abstracting communication in distributed agent-based systems. In: Proceedings of the 16th European Conference on Object-Oriented Programming. (2002)
19. Willmott, S., Constantinescu, I., Calisti, M.: Multilingual agents: Ontologies, languages and abstractions. In: Proceedings of First International Workshop on Ontologies in Agent Systems. (2001)
20. Payne, T., Singh, R., Sycara, K.: Communicating agents in open multi-agent systems. In: First GSFC/JPL Workshop on Radical Agent Concepts (WRAC). (2002)
21. Nodine, M.H., Unruh, A.: Facilitating open communication in agent systems: The InfoSleuth infrastructure. In: Agent Theories, Architectures, and Languages. (1997) 281–295