Using Ontologies in the Development of an Innovating System for Elderly People Tele-Assistance[1]

M. I. Bagüés, J. Bermúdez, A. Illarramendi, A. Tablado and A. Goñi.

University of the Basque Country (UPV/EHU)
LSI Department. Donostia-San Sebastián. Spain
jipileca@si.ehu.es
http://siul02.si.ehu.es/

We have used most recent advances in the fields of artificial intelligence (knowledge representation), mobile computing and networking to develop the AINGERU system that provides a new kind of tele-assistance service for elderly people. The new features incorporated over the traditional tele-assistance are: high quality monitoring, anywhere and anytime. In this paper we focus on the high quality aspect and, more precisely, we present the role that two specific ontologies play and the advantages that they provide to the system. One ontology describes domain knowledge and the other one describes the operational model of the system.

1. Introduction

Nowadays, ontologies are used in many application domains and their usefulness is widely accepted. The goal of this paper is to present the interest of using two specific ontologies that describe knowledge of different nature, within an application called AINGERU. This system uses agent technology, semantic web technology and Personal Digital Assistant (PDA) with wireless communications, to provide high quality assistance to elderly people, giving them more freedom and protection.

Many recent surveys agree with the following statement: "the population of the first world is aging". Traditional Social Services could well be overwhelmed within 30 years. Another important issue is that elderly people are becoming more and more independent. As medical science advances, people can live alone with better health up to a very advanced age. Therefore, in order to let elderly people live in their own homes leading their normal life, while, at the same time taking care of them, special purpose services are required. Up to now, most of tele assistance services offered are based on specific hardware (e.g. [1,2,3]), but all of them are constrained by their limited coverage and functionality. Due to this, they do not fulfill the aim of a *high-quality*, *anywhere* and *anytime* assistance.

Nevertheless, apart from supporting the functionalities provided by present tele assistance services, AINGERU also: offers an *active assistance* by using agents that

[1] This work was mainly supported by the University of the Basque Country and Diputación Foral de Gipuzkoa

behave in face of anomalous situations; offers an *anywhere* and *anytime* assistance by using wireless communications and PDAs; and, allows to *monitor vital signs* by using sensors that capture the values of those signus and feed a decision support system that analyzes them and generates an alarm when necessary.

The main elements that enable AINGERU to offer those functionalities are:

- A set of PDAs, one for each user, which can be carried everywhere by these users and which support wireless communication with other devices such as sensors and computer servers.
- A set of computers located at different centers that mainly take part of a sanitary network.
- A set of specialized agents that work autonomously accomplishing tasks that have been assigned to them and that are parts of the global application. For example, an agent called Localization Agent is in charge of knowing where the user is at every moment, while other agents could ask him this information whenever needed.
- Two logic-based ontologies that allow reasoning on them, and web services that support enquires about user data stored at the PDA.

Although in the following section we show some features of the agents and PDAs, the main focus of this paper is on the two above-mentioned ontologies. Notice that the development and use of these two ontologies is a novelty that AINGERU brings with respect to other tele assistance services. Both ontologies are described using semantic web technology [2].

The purpose of the first ontology, called *MedOnt*, is to describe the different situations in which a medical alarm has to be activated. Hence, in this ontology not only are the different symptoms that a user can have described, with respect to Vital Constants that several sensors can monitor, but also the usual illnesses that elderly people suffer from. Moreover, this ontology can be customized for every user. That is, this ontology can be easily adapted to different users depending on their own situation. This ontology is being developed by experts (so far we only have a small prototype).

The purpose of the second ontology, called *OperOnt,* is to describe the operational model of AINGERU. For example, in this ontology we describe messages for the communication among agents in AINGERU. The concepts in this ontology are defined independently from any agent system implementation, so that it helps interoperability among agents without pre-defined agreements. At the same time, the *OperOnt* ontology describes contextual information that several agents are able to share. This ontology is easily extensible as the functionality of AINGERU increases.

The main advantages of using the above-mentioned ontologies in our AINGERU system are:

- The ontologies allow sharing knowledge (in our case, medical and operational) with other systems, so they favor the interoperability of AINGERU with other systems.

---

[2] Right now DAML+OIL [4] is used to describe the ontologies, but there is a direct translation to OWL [5], the W3C web ontology language.

- The *OperOnt* ontology allows the interoperation of independently generated agents without pre-establishing specific protocols. The communication among agents is at the semantic level.
- Reasoning with the *MedOnt* ontology allows the system to activate alarms for many different situations that can happen to the users.

In the rest of this paper, first we briefly describe the global framework of AINGERU. Then, we show the features of the *MedOnt* and *OperOnt* ontologies. We continue presenting a trace of the system at work and some related works. We finish with some conclusions.

2. Global Architecture of AINGERU

The goal of this section is to present the main features of the AINGERU architecture, which appears in Fig.1, from two different perspectives. First, from the physical perspective, i.e., the components that take part in the architecture and the type of communication among them. Next from the software perspective, i.e., the main software elements – specialist agents, ontologies, reasoner and web services– that operate to accomplish the pursued goal of AINGERU

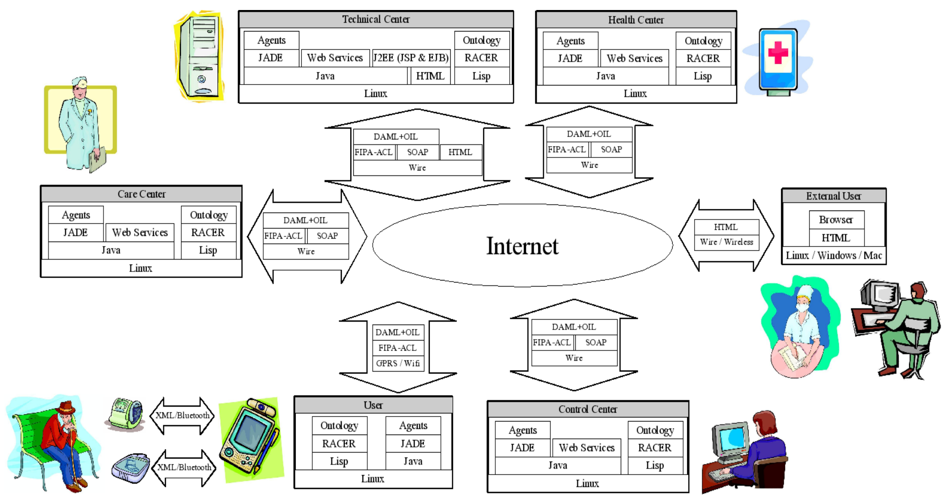


Fig. 1. Global architecture of AINGERU

2.1. Physical Perspective of AINGERU

As can be observed in Fig. 1, there are five different types of components, namely: PDAs, Control Center, Care Center, Health Center and Technical Center. AINGERU supports the distributed nature of the application domain.

Personal Digital Assistant PDA

Each person monitored by AINGERU carries a PDA. The PDA is a central element in AINGERU architecture and its main goal is twofold, first to monitor the user and then, when special circumstances require it, to be the link between the person and the center (Control Center) that is also responsible for monitoring her/him. Due to its reduced size a PDA can be carried anywhere. Moreover, the technical features that PDAs nowadays support, allow us to run the software application of AINGERU in them. In the following paragraphs we mention three basic functionalities of PDAs: manual activation of an alarm, automatic activation of an alarm and agenda services.

Manual activation of an alarm. Devices nowadays (those mentioned in the introduction) allow people to activate an alarm from their home, mainly by pressing a button. AINGERU also supports this functionality by providing the user with the interface shown in Fig. 2. When the user feels bad, or something is happening in his environment and he wants to notify it, he presses the button that appears in the interface and an alarm is activated in the Control Center. Notice that in the case of AINGERU this functionality is not constrained to users' homes, as it occurs nowadays.

Fig. 2. One of the interfaces of the AINGERU system

Automatic activation of an alarm. Aside from the previous basic functionality an added value which AINGERU contributes to is the possibility of controlling in situ the vital constants of a person in order to monitor risk situations. The PDA receives data sent by sensors (even by wireless sensors) related to pulse, mobility, etc., and analyzes them in order to activate an alarm when an anomalous situation is being detected.

Agenda services. Apart from the two mentioned functionalities, the users' PDA provides them with classical agenda services such as remembering when they have to visit the doctor, when they have to take their medicines, which appointments they have, etc.

Control Center

Control Centers are the centers in charge of monitoring people. Each Control Center hosts a computer called Control Center. The number of Control Centers depend on how many users must be monitored. The main goal of a Control Center is to react in the presence of user alarms and to take the adequate actions.

Care Center

Care Centers are the public health centers for primary assistance. Each Care Center hosts one or more computers. Part of the AINGERU application must run in one of those computers (called Care Center) in order to provide AINGERU users with new functionalities such as: accessibility of physicians to data stored in the user PDA, direct insertion of medical appointments in the PDA, etc.

Health Center

Health Centers correspond to hospitals. In the presence of an alarm, the Control Center could decide that the user must be moved to a hospital. In that case, the Control Center would send information about the user to the hospital so they could be prepared when receiving the user. For this reason, part of the AINGERU application must also run at one Health Center computer (called Health Center) without interfering with existing applications.

Technical Center

The goal of this center is the development and support of the AINGERU application. It hosts a computer called Technical Center. It is in charge of providing the PDA users with new software releases, new services, and so on.

Concerning communication aspects among the components, in figure 1 we can observe the different levels that are used. At the physical level the wired or wireless communication appears. At the transport level, FIPA is used for inter agent communications and SOAP for web services. At the application level, agents and web services communicate through ontologies described in DAML+OIL.

2.2. Software Perspective of AINGERU

In Fig. 1 we can observe the main software elements that take part of AINGERU: ontologies, web services and agents. In this subsection we briefly present web services and agents because ontologies are explained in more detail in the following sections.

Web Services

In AINGERU we have developed web services that export part of agent knowledge and allow desktop applications to dive into the agency world. This possibility, not taken into account in the tele-assistance applications nowadays, is very interesting, and opens new opportunities for future extensions of AINGERU. Examples of these web services are:

- *Vital Constants:* A web service exported by the Care Center. It provides information about user vital constants in real-time. Authorized agents (physicians, for example) can obtain data about the current values of the sensors that monitor the user.

- *Location:* A web service exported by the Technical Center. It provides information about the location of the user. Location information is managed by the Location Agent residing in the User Agency within the PDA.
- *Appointments:* Web services exported by the Care Center and Technical Center. They manage the user's appointments (with physicians, with relatives, etc). The User Agency will alert the user on time.
- *Medical Records:* Web service exported by the Care Center. It allows physicians to obtain medical records of the users.

These services can be invoked via SOAP. But we have also developed an HTML front-end to those web services. This facility will allow that anybody that has a web browser installed on their computers to access those web services.

Agents

We decided to use agent technology in AINGERU because features that agents provide are vital in our context. Among those features we point out the following: agents are autonomous, reactive and can be mobile. The autonomy feature allows us to distribute the global functionality of the AINGERU system among different agents, each agent being responsible for performing a specific task and controlling its reactions independently of others. The reactiveness feature allows the development of an active assistance which responds to changes in the user health conditions. Last but not least, mobility allows agents to travel from the user's PDA to Care Center, Health Center or Control Center computers and vice versa, in order to perform certain tasks locally, saving wireless communications costs.

Furthermore, a set of agents can cooperate to accomplish a task, and this task can be a step to achieve a goal. For example, two agents can cooperate to check a person's pulse: the *Bluetooth Agent*, which knows how to deal with bluetooth wireless technology, and the *Pulse Agent*, which manages specific knowledge related to pulse. Moreover, checking the pulse may be only a step in vital constants control.

Among the different agents that we have developed in AINGERU, we only mention here the details of two of them: *Majordomo* and *Conditions Checker* (see [6] for more information).

The *Majordomo Agent* is located at the user's PDA and its goal is to be the link between the user and the AINGERU system. It shows information on the screen or tells it through the speaker, and it gets user's response by handwriting recognition on screen, speech recognition or pressing buttons on the interface. This agent can be customized for different users.

The *Condition Checker Agent* is also located at the user's PDA and its goal is to monitor user health values. To accomplish its goal this agent makes use of the *MedOnt* ontology explained in section 3. The behavior of this agent is as follows: when it receives data, captured by sensors, it pushes those values into the reasoning system as values filling properties associated to concepts in the *MedOnt* ontology and observes the inferences made by the reasoner. If an instance is recognized into the `Alarm` concept the Condition Checker Agent activates an alarm and sends it to the Control Center. This agent, the *MedOnt* ontology and its supporting reasoner, constitute the Decision Support System that permits AINGERU to provide its users with a high quality tele-assistance service.

After trying out some implementations for agents, we selected the JADE system because it was the one that most accurately fulfilled our requirements. Key features that affected our decision are:

- Standards-compliant: JADE is compliant with the latest FIPA standards [7]. We use FIPA as a transport level standard protocol. Above that, agents communicate at a semantic level provided by the *OperOnt* ontology.
- Lightweight: because some agencies will be running in PDAs, we need a platform that is not very heavyweighted. JADE has versions that can run on J2ME/MIDP complaint devices, like phones and watches.
- Generality: it allows us to use different transport drivers to carry messages among agencies. For example, there are drivers that use CORBA, HTTP or even SMTP (mail protocol).
- Security: in addition to establishing secure channels based on SSL communications, JADE allows to sign messages by binding an identity to every message. This prevents unauthorized agents killing other agents. Furthermore, message integrity is assured, preventing message tampering.

There are other systems that also promote applying agents in Health Care [8,9,10,11]. However, as far as we know, they are still in their initial development and they do not put special emphasis on combining the agent technology with the use of PDAs.

3. Medical Ontology (*MedOnt*)

As we have mentioned in the introduction, our AINGERU system, apart from allowing anywhere and anytime monitoring, has the goal of providing a high quality monitoring. This means that AINGERU not only will generate an alarm when the user requires it, but also when the system autonomously detects anomalous situations. Thus, its behavior is reactive when the user's health is in danger. In our opinion this last feature is a step forward in the field of tele-assistance applications.

In order to accomplish the goal of detecting anomalous situations, two components are necessary: first, sensors that capture users' health values such as temperature, pulse, etc. and then, a knowledge based system that analyzes the captured values. Nowadays there is a tremendous interest, from the research point of view as well as from the commercial point of view, in developing wireless intelligent sensors that capture biological signal data and transmit these data using bluetooth technology. Aspects relative to these sensors are beyond the scope of this paper, but we believe that they will be added to the AINGERU system in the near future.

Concerning the development of the knowledge based system to represent data sent by the sensors, we think that the ontology based approach, following the direction promoted by the Semantic Web vision, is far more adequate than a rule based representation approach. After all, the latter presents the following disadvantages: it tends to be less open, it is biased towards case by case analysis, and its validity is more difficult to reason about. By contrast, the ontology based approach consists of describing concepts, at a semantic level using well founded operators. In our case, concepts describe states of illnesses that elderly people can suffer as well as states generated by the values captured by the sensors. With respect to our *MedOnt* ontology we follow the approach of GALEN [12] (which, in turn, is directly related to the well-known SNOMED [13] medical terminology) although we do not use the GALEN Representation and Integration Language (GRAIL) [14]. We decided to use DAML+OIL (an ontology representation language from the Semantic Web forum, with a direct translation to the recently launched OWL) to take advantage of semantic web technology and its richer primitives for concept descriptions. Furthermore, DAML+OIL allows the description of some datatypes, so we are able to deal with properties with integer values and with structured values such as HL7 [15] rows. Moreover, for the reasoning process we selected the RACER [16] system, which implements a highly optimized tableau calculus for a very expressive description logic that covers all the DAML+OIL primitives we need; in particular, it deals with reasoning about individuals of concepts, which is crucial in our framework.

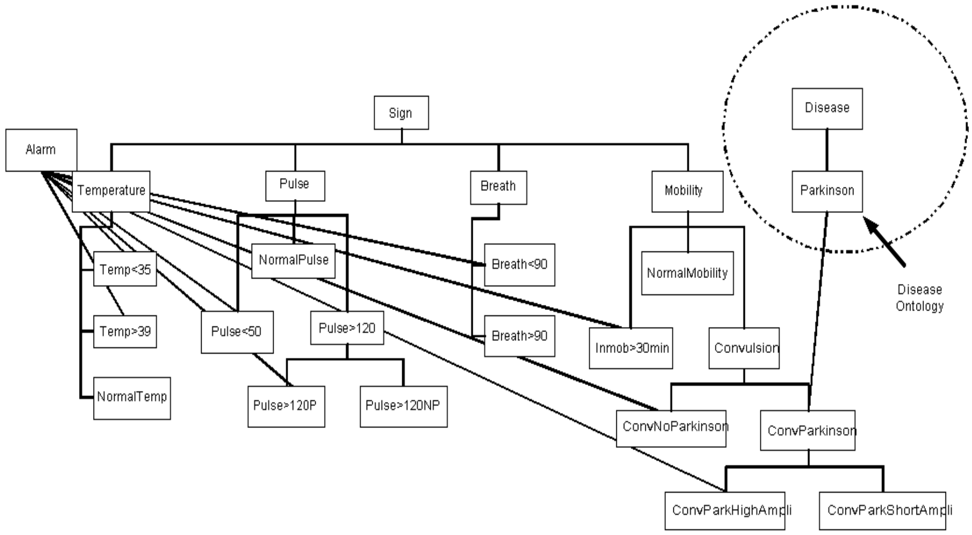


Fig. 3. Graphic representation of a fragment of the *MedOnt* ontology

The *MedOnt* ontology is being built by physicians who are experts in the care of elderly people. So far, to build the AINGERU prototype, we deal with a “toy” *MedOnt* ontology that we have built with the collaboration of physicians (see Fig. 3 and Fig. 5). For example, in this ontology we can observe the `Pulse<50` concept that describes the bradycardia situation and which is related to the `Pulse` signs. Concepts in this ontology fragment are described according to the values captured by different sensors (temperature, pulse, etc.). There are concepts describing normal situations such as `NormalPulse`, and others describing anomalous situations, such as `Pulse>120P`, that are recognized as (*subsumed by*) an `Alarm` situation.

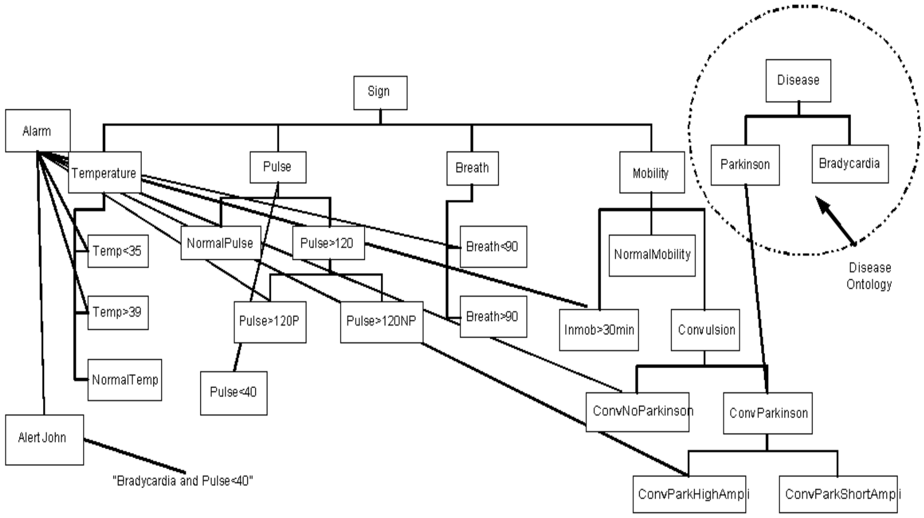


Fig. 4. Graphic Representation of a fragment of a Personalized *MedOnt*

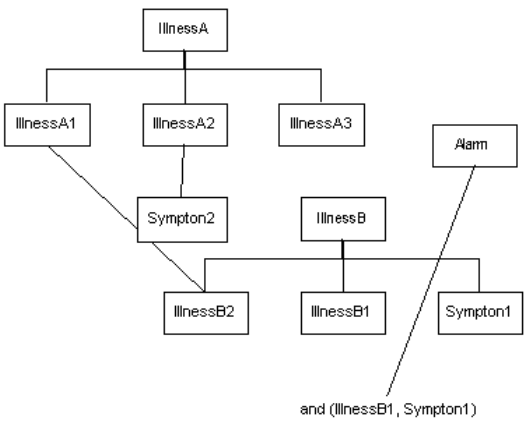
In Fig. 4 the `AlertJohn` concept illustrates the desirable customization of the *MedOnt* ontology loaded into each user's PDA (`AlertJohn` for the user John). The `AlertJohn` concept is a kind of `Alarm` that describes all the potential anomalous situations of the user of this particular PDA. We consider customization a very interesting feature because every single person has peculiarities added to general symptoms; for example, the state of John who suffers `Bradycardia` is not in an alarm situation if the pulse is below 50 but above 40: `AlertJohn` is, in fact, activated if the pulse is below 40. Notice that in the customization process the concept `Pulse<50` has been removed.

There are also some concepts in the *MedOnt* ontology that describe states combining `Sign`s detected by sensors with a state of disease. For example, the concept `Convulsion` describes the state when the Mobility Sensor reports shaking over a certain threshold and `ConvParkinson` describes the state of the kind of `Convulsion` of a person who suffers from `Parkinson`'s disease. Notice that `ConvParkinson` is not considered a state of alarm. By contrast, `ConvNoParkinson` describes a kind of `Convulsion` of people not suffering from Parkinson's disease, and this concept is considered a kind of `Alarm` because the person is possibly suffering from an epilepsy episode.

```
<daml:Class rdf:about="file:/C:/Aingeru/OntoMed.daml#Breath">
  <rdfs:subClassOf>
    <daml:Class rdf:about="file:/C:/Aingeru/OntoMed.daml#Sign"/>
  </rdfs:subClassOf>
</daml:Class>
<daml:Class rdf:about="file:/C:/Aingeru/OntoMed.daml#Breath&lt;90">
  <rdfs:subClassOf>
    <daml:Class rdf:about="file:/C:/Aingeru/OntoMed.daml#Breath"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Class rdf:about="file:/C:/Aingeru/OntoMed.daml#Alarm"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="file:/C:/Aingeru/OntoMed.daml#value"/>
      <daml:toClass rdf:resource="http://siul02.si.ehu.es/Aingeru/Schemas#BreathMin90"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

Fig. 5. Definition of a fragment of the *MedOnt* ontology in DAML+OIL

Nevertheless, we want to stress that *MedOnt* is only a toy ontology and in the near future we hope to develop a real one that allows making smart reasonings, such as the one described in the following abstract example.




```
class-def IllnessA
class-def IllnessA1
        subclass-of IllnessA
class-def IllnessA2
        subclass-of IllnessA
class-def IllnessA3
        subclass-of IllnessA
disjoint-covered IllnessA
        by IllnessA1 IllnessA2 IllnessA3
class-def IllnessB
class-def IllnessB1
        subclass-of IllnessB
class-def IllnessB2
        subclass-of IllnessB IllnessA1
disjoint-covered IllnessB
        by IllnessB1 IllnessB2
class-def Symptom1
        subclass-of IllnessB
class-def Symptom2
        subclass-of IllnessA2
class-def Alarm
and ( IllnessB1 ,Symptom1)
        subclass-of Alarm
```

Fig. 6. Graphic representation of a fragment of an ontology and its specification using an abstract syntax.

Assume a conceptualization as shown in Fig. 6, describing different kinds of illnesses and their relationships. Assume that there are also two symptoms - `Symptom1` and `Symptom2`- whose conceptualization –accorded by the ontology designers- classifies them as subsumees of `IllnessB` and `IllnessA2`, respectively (i.e., it is admitted by ontology designers that if `Symptom1` is recognized about a person, then that person suffers from `IllnessB`, and similarly for `Symptom2` and `IllnessA2`). In addition, it is described as an `Alarm` when a person suffers from `IllnessB1` and `Symptom1`.

Now, assume that it is recognized that a person suffers from Symptom1 and Symptom2. Then, the reasoning mechanism will infer that this person is in an Alarm situation. The inferencing process works this way: Symptom1 implies that the person suffers from IllnessB, although what specific subclass is still uncertain -IllnessB1 or IllnessB2-. On the other hand, Symptom2 implies IllnessA2. Moreover, IllnessA1 is discarded due to the specification of the IllnessA concept as the pairwise disjoint union of the IllnessA1, IllnessA2 and IllnessA3 concepts (notice the disjoint-covered operator in Fig. 6). Then, IllnessB2 -being subclass of IllnessA1- must be also discarded. Finally, it is inferred that IllnessB must specifically be an IllnessB1 and, consequently, there is an Alarm.

This is the kind of reasoning provided by the description logic underlying DAML+OIL.

4. Operational Ontology (*OperOnt*)

As mentioned in section 2.2, agents play a very important role in the AINGERU system. Those agents are implemented using the JADE agent system; nevertheless, taking into account that interoperability with other systems is a goal for AINGERU, we have defined an ontology, called *OperOnt*, for the description of concepts that allow the operation of agents at a semantic level. That means that the ontology includes descriptions of agents, messages, and subjects of the messages.

Communication among agents takes place by sending messages. A message has properties. For example a BluetoothSensorQuery message has the following properties: OperOnt:subject, OperOnt:priority, OperOnt:ident, OperOnt:sender, OperOnt:receiver and OperOnt:sendingNumber

The concepts and properties in *OperOnt* offer a well founded common understanding of terms for the interaction of agents. This approach favors, on the one hand, interoperation since new agents wishing to interact with the system only need to browse the corresponding ontology, and on the other hand, evolution because properties of messages can be modified without altering the agents that work with those messages.

For example, if an agent wants Bluetooth Sensor information, it only has to create an instance of the Query concept filled with the property OperOnt:subject= "BluetoothSensor", and push it into the reasoning system. The reasoning system will infer that the message to be sent is a BluetoothSensorQuery and will inform the agent about the properties the message must fulfill. Similarly, if an agent receives a MedicalAppointmentModify message it will push it into the reasoning system, which will infer that somebody is making a Request (see Fig. 9) to modify a Medical Appointment, as well as which properties the message has.

The *OperOnt* ontology is basically divided into three interrelated areas (in URL http://siul02.si.ehu.es/Aingeru/OperOnt.owl can be seen the specification of this ontology): (i) The actors who interact using messages, (ii) the subjects of the messages, and (iii) the functionality of the messages. Next we will outline each area, but we want to stress that there are axioms in the ontology that describe their interrelationships. Furthermore, it is worth mentioning that the formalism used in this approach together with the semantic web technology enable the proper integration of new areas into this ontology.

4.1. Actors

We divide this area (see Fig. 7) into human agents, software agents and AINGERU web services. Human agents are classified into two groups: on one hand AINGERU users and in the other hand all those that form part of the system (sanitary and non-sanitary).

Software agents are described taking into account their location and their goals (for instance, whether they work in a PDA or in a computer, if they are attending a sensor or interacting with an ontology, and so on). Web services are described on the basis of their functionality.

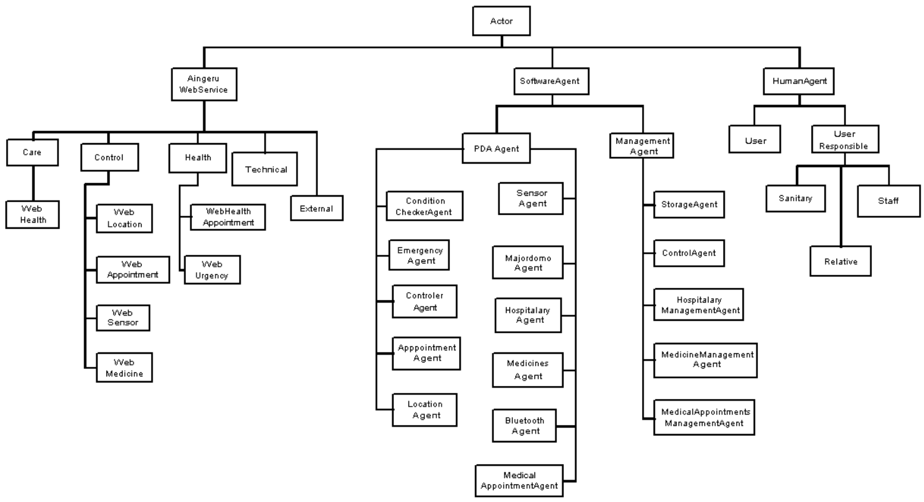


Fig. 7. Actors area

4.2. Subject of the messages

This area describes concepts about the subject or the topic on which the message is centered (see Fig. 8). For example, `InputOutput`, `Location`, `Urgency`, `Emergency`, `Sensor`, `HospitalSubject`, `Appointment` and `Medicine` describe different kinds of subject.

More specialized subjects are defined taking into account their context (for example origin, destination, etc.). For instance, `GeneralSensor` is related to sensors that the user carries, `BluetoothSensor` is related to the bluetooth transmission performed by the sensors and `DumpSensor` is related to historical information saved in sensor agents.

This conceptualization describes subjects of messages in order to interpret their contents properly.

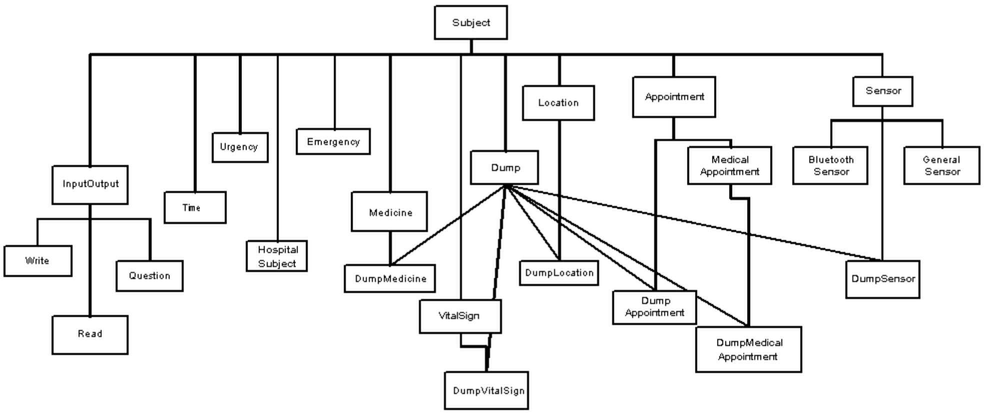


Fig. 8. Subjects area

4.3. Functionality of the messages

Considering the standards-compliant principle in AINGERU, we have included descriptions of messages according to their functionality in FIPA protocols (see Fig. 9).

Message descriptions are based on three FIPA[3] protocols: FIPA Request Interaction Protocol, FIPA Subscribe Interaction Protocol and FIPA Query Interaction Protocol. Types of messages that appear in *OperOnt* are: `Agree`, `Cancel`, `Failure`, `Refuse`, `Query`, `Subscribe` and `Inform`.

Another feature that differentiates messages is the particular subject of the message content. For example, if we have two messages whose functionality is `Query-ref`, they are different kinds of messages if they differ in the kind of subject they request (for instance, if one requests a `Location`, and the other requests some `Medicine`).

With these examples we also show the relationships among concepts presented in different areas. Typically, conceptualization in one area is used as qualification for concepts in another area, i.e., different concepts in one area serve as a domain or a range for properties in the other.

[3] At the current state of AINGERU we use those three protocols. However, more could be considered if necessary.

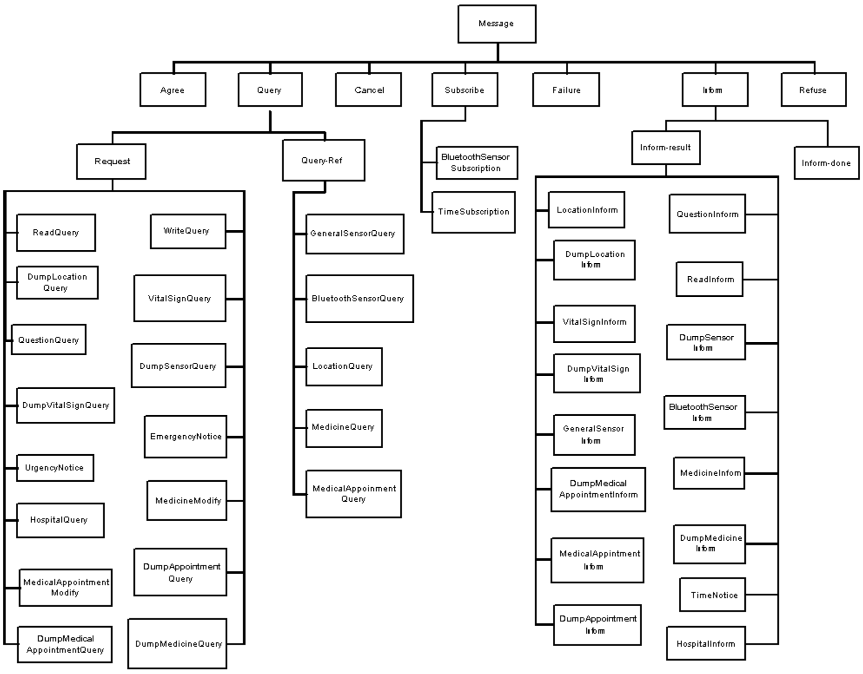


Fig. 9. Functionality of the messages area

5. Trace of AINGERU at work

For the construction of the AINGERU system we have used an object oriented development process based on UML (Unified Modelling Language) [17]. In this section we show a trace of AINGERU system at work, corresponding to the use case represented in Fig. 10. This use case considers the situation in which the alarm is activated automatically.

The *Condition Checker Agent* periodically receives information from different Sensor Agents that exist in the PDA (every Sensor Agent controls the information obtained by one of the sensors that the user carries). When the *Condition Checker Agent* receives this information (for example pulse of 125), it pushes `s <MedOnt:value> 125` into the reasoning system and observes the inferences made by it. If `s` is recognized as an instance of the `Alarm` concept, the *Condition Checker Agent* asserts the following statements to create a new message:

```
m : OperOnt:Query[4]; ur : OperOnt:Urgency;

m <OperOnt:subject> ur;
```

[4] We use abstract syntax

Then the reasoning system will infer that `m` is an instance of the `OperOnt:AlarmNotice` concept.

```
m : OperOnt:AlarmNotice;
```

Next the *Condition Checker Agent* ask the Knowledge Base System to retrieve the properties associated to `OperOnt:AlarmNotice` (in our case this is done using RICE [18]). It will receive the following list of properties:

```
{OperOnt:ident; OperOnt:subject; OperOnt:theSender;
OperOnt:theReceiver; OperOnt:tryNumber}
```

With this information the *Condition Checker Agent* can create the `AlarmNotice` message to be sent to the *Emergency Agent.*

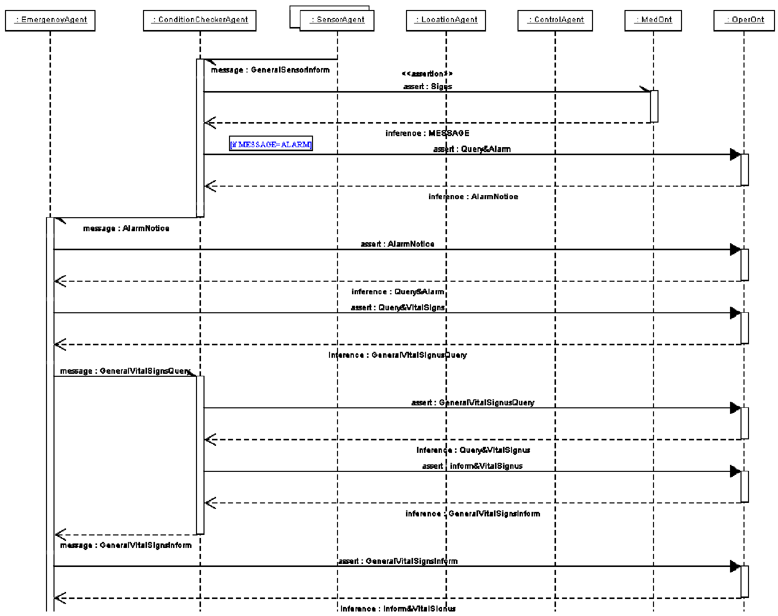


Fig. 10. UML Sequence Diagram of some steps of the use case “Automatic Alarm Activation”

Then, the *Emergency Agent* asks the *Condition Checker Agent* for all the vital constants, and the *Location Agent* for the location of the user.

As soon as the *Emergency Agent* receives all the information, it creates the `EmergencyNotice` message (following the above-mentioned steps for creating the message using the *OperOnt* ontology). This message appears in Fig. 11.

```
<OperOnt:EmergencyNotice>
        <OperOnt:ident>
                M1234-A
        </OperOnt:ident>
        <OperOnt:theSender>
                <OperOnt:EmergencyAgent />
        </OperOnt:theSender>
        <OperOnt:theReceiver>
                <OperOnt:ControlAgent />
        </OperOnt:theReceiver>
        <OperOnt:tryNumber>
                1
        </OperOnt:tryNumber>
        <OperOnt:subject>
                <OperOnt:TheVitalSignus>
                        MSH|^~\&|EMERGENCY AGENT||CONTROL AGENT|200229091030|| ORF^R04|RCVP001|P|2.3.1<cr>
                        MSA|AA| PCVP001<cr>
                        QRD|200229091030|R|||QCV0001|||20|RD|123456789|RES<cr>
                        QRF|EMERGENCY AGENT||200229091030<cr>
                        PID|1||11234234-N||RODRIGUEZ^JUAN|JIMENEZ||M|||||943565656 <cr>
                        OBR|4|P8756^OE|N2345^NR|3000.02^VITAL SIGNS|||200229091028|||SENS_PULSO^SENS_COR |N<cr>
                        OBX|1|ST|8462-4^INTRAVASCULAR DIASTOLIC:PRES:^LN||90|mm(hg)|60-90||||F<cr>
                        OBX|2|ST|8479-8^INTRAVASCULAR SYSTOLIC:PRES:^LN||120|mm(hg)|100-160||||F<cr>
                        OBX|3|ST|8478-0^INTRAVASCULAR MEAN:PRES:^LN||100|mm(hg)|80-120|N|||F<cr>
                        OBX|4|ST|8867-4^HEART BEAT:NRAT:^LN||125|/min|60-100|HH|||F<cr>
                </OperOnt:TheVitalSignus>
                <OperOnt:EmergencyType>
                        Medical Emergency
                </OperOnt:EmergencyType>
                <OperOnt:TheLocation>
                        43°18'26",-2°0'41"
                </OperOnt:TheLocation>
        </OperOnt:subject>
</OperOnt:EmergencyNotice>
```


Fig. 11. EmergencyNotice message in DAML+OIL

Finally, the message in Fig. 11 is sent to the *Control Agent* (located at the Control Center). Then, the Control Center decides what to do depending on the information received. Moreover, as it is a medical emergency (since the alarm has been activated by the *Condition Checker Agent*) an ambulance will be sent to wherever the user is.

6. Related works

Works related to AINGERU may be classified into two major groups: works whose aim is to build tele-assistance applications for elderly people, and works that advocate for the use of ontologies in the Pervasive Ubiquitous Computing Environments.

We also classify tele-assistance applications in two groups. In the first group we include those systems that provide limited coverage, such as existing tele-alarms. The main features of these systems are the following: they use wired phone communications to contact Social Services, their coverage is restricted usually to the user's home and their activation is triggered by the user, generally using a button. Therefore, they do not support *anywhere* and *anytime* assistance. In the second group we include more advanced systems. The coverage provided by these systems is broader: they use PDAs and take advantage of wireless communication. They provide *anywhere* and *anytime* assistance, but they are not *reactive*. PDAs are used as intermediary elements and their goal is merely reduced to the transmition of data from sensors to a central computer where data analysis is made. They do not take advantage of the ability of the PDAs to carry out a certain pre-analysis before sending data to the central computer. Notice that wireless communications are slow, expensive and unstable so, analysis made in the PDA can save costs and detect anomalies earlier. Examples in this group are doc@HOME [1] , Sensatex [19] , SILC [2] or TeleMediCare [3]. AINGERU goes one step further by providing not only anywhere and anytime assistance using wireless communication, but also a high quality assistance, mainly due to the use of agents and semantic web technology.

With respect to the use of ontologies in a pervasive computing environment, we point out two works [20,21]. The first paper [20] presents the features of the CoBrA ontology described in OWL, that models the basic concepts of people, agents, places and presentation events in intelligent meeting room environments. Two main features differentiate our work from their approach: first, in our work the reasoning is made at the mobile device, while in their case it is made at the stationary servers. While it is true that doing the reasoning process at the PDA has not been an easy task, we have verified its feasibility and in our opinion the advantages that provide the local reasoning justify its use. The second difference corresponds to *OperOnt* ontology. In CoBrA ontology they do not consider the type of terms and the goal pursued by our *OperOnt* ontology. In the second paper [21], they show how the use of ontologies can help overcome three major issues that confront the development and deployment of Pervasive Computing Environments (discovery and matchmaking, inter-operability between different entities and context-awareness). Although we agree with most of their conclusions, our work is centered in one specific application domain.

7. Conclusions

To develop new social and sanitary programs, as well as new software and hardware systems that help elderly people to increase their quality of life, is a challenge nowadays posed in developed countries.

In this paper we have presented our system, called AINGERU, that gives one step forward in that direction of improving the quality of life, by allowing the monitoring of people anywhere and anytime. Moreover, this monitoring is reactive, i.e., not only is an alarm activated when the user requires it, but also when the system detects an anomalous situation. In order to perform the detection process, the system makes an extensive use of an application domain ontology, *MedOnt*, that describes states outlined by vital constant values that must be monitored, as well as diseases that elderly people can suffer from. Another contribution of the system is the development and management of an operational ontology, *OperOnt*, that permits agents that take part in the system to communicate at a semantic level.

References

[1] doc@HOME, The bridge to care. http://www.docobo.com, (2003, July).
[2] Supporting Independently Living Citizens. http://www.fortec.tuwien.ac.at/silcweb/SILC.htm, (2002, July).
[3] The TelemediCare project. http://www.telemedicare.net, (2003, July).

[4] Connolly,D., Van Harmelen, F., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A. : DAML+OIL (March 2001) Reference Description. http://www.w3.org/TR/daml+oil-reference (2003, July).

[5] OWL Web Ontology Language Guide. http://www.w3.org/2001/sw/WebOnt/guide-src/Guide.html. (2003, July).

[6] Tablado, A., Illarramendi, A., Bermúdez, J., Goñi, A.: "Intelligent monitoring of elderly people". in *4th Annual IEEE Conference on Information Technology Applications in Biomedicine*, 2003, pp 78-81.

[7] FIPA. Foundation for Intelligent Physical Agents. http://www.fipa.org/specifications/index.html. (2003, July).

[8] Mouratidis, H., Manson, G., Giorgini, P., and Philp, I.: "Modelling an agent-based integrated health and social care information system for older people". In *15th European Conference on Artificial Intelligence*, 22 July 2002.

[9] Mabry, S. L., Kollmansberger, S. J., Etters, T., and Jones, K. L.: "IM-Agents for patient monitoring and diagnostics". In *15th European Conference on Artificial Intelligence*, 22 July 2002.

[10] Moreno, A. and Isern, D.: Offering agent-based medical services within the AgentCities project. In *15th European Conference on Artificial Intelligence*, 22 July 2002.

[11] Camarinha-Matos, L. M. and Afsarmanesh, H.: Virtual Communities and Elderly Support. *Advances in Automation, Multimedia and Video System, and Modern Computer Science*, pages 279-284, September 2001.

[12] OpenGALEN: Making the impossible very difficult. http://www.opengalen.org/. (2003, July).

[13] SNOMED International. http://www.snomed.org/. (2003, July).

[14] Rector, A. , Bechhofer, S. et. al.: "The GRAIL concept modelling language for medical terminology" Artificial Intelligence in Medicine, 9: 139-171 ,1997.

[15] HL7 Version 2.3 Implementation Guide. http://www.hl7.org/Special/IG/final.pdf (2003, July).

[16] Haarslev, V., Möller, R.: RACER: Renamed ABox and Concept Expression Reasoner. http://www.fh-wedel.de/~mo/racer/ (2003, July).

[17] UML Resource page. http://www.omg.org/uml/ (2003, July).

[18] Cornet, R. RICE: Racer Interactive Client Environment http://www.b1g-systems.com/ronald/rice/ (2003, July)

[19] Sensatex. http://www.sensatex.com. (2003,July).

[20] Chen, H., Finin, T. and Joshi, A.: An Ontology for Context-Aware Pervasive Computing Environments. Eighteenth International Joint Conference on Artificial Intelligence Workshop on Information Integration on the Web(IIWeb-03) August 9-10, 2003 Acapulco Mexico.

[21] Ranganathan, A., McGrath, R. E., Campbell, R. H., Mickunas, M. D.: Ontologies in a Pervasive Computing Environment. Eighteenth International Joint Conference On Artificial Intelligence Workshop on Information Integration on the Web(IIWeb-03) August 9-10, 2003 Acapulco Mexico.