Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals of DAAAM International

PATIENT MONITORING WITH AGENTS AND SEMANTIC WEB

CRISTESCU, S[orin] - A[lexandru] & MOLDOVEANU, F[lorica]

Abstract: *Current patient monitoring systems are typically connected uncomfortably to the patient's body and to other devices, with which they communica*
e. They are limited in the way they exchange information, as they can only communicate to the systems they are connected with, usually using proprietary protocols.
With the elderly population growing all over the world, the need for efficient patient monitoring systems increases too. The authors of this paper propose a way to employ existing technologies in order to improve the quality of the medical act, in particular regarding remote patient monitoring systems.
Key words: *patient monitoring, multiagent, semantic web*

1. INTRODUCTION

Remote patient monitoring systems have been around for some years now. However, they are limited both in their capabilities and in the way they communicate with other devices. The monitoring systems are typically connected uncomfortably to the patient's body and to other devices, to which they transmit vital signals. Even if they use wireless communication, their interoperability is limited to gathering some data from the patient and communicating it to certain hardwired devices. They are unable to interact to other devices and they are triggered by patient or doctor intervention.

This paper presents a more flexible framework for remote patient monitoring, leveraged by multiagent systems, ontologies and semantic web.

Another limitation of the current monitoring systems is their incapacity to work without input from a human operator or device. In contrast, software agents are autonomous, so they don't need triggers from their users, but they rather react autonomously to stimuli from the environment.

The authors of this paper propose a platform that combines the autonomy and interoperability of multiagent systems with the legacy-enabling web services in an effort to enhance the interactions among the stakeholders of the medical domain, specifically regarding the patient monitoring process. As we'll see in the next sections, at the core of the platform are ontologies and ontology mapping systems.

2. MOTIVATION

With the worldwide elderly population growing fast, the need for efficient patient monitoring systems increases too. As mentioned in the previous section, the current monitoring systems suffer from important limitations.

Let's assume the following scenario: *"John, a 70 year old grandfather is taking his 2 year old grandson for a walk in the park. Suddenly John feels bad and collapses to the ground. There is nobody at that hour in the park. Luckily, John wears his intelligent watch, which, by the means of a software agent and wireless Internet access, alerts the emergency service. Namely, it sends the patient's personal data, his symptoms, his medical record as well as his current location. Knowing the patient's location, the emergency service looks up the closest available ambulance, by informing the ambulance dispatcher agents in the neighborhood. In less than 5 minutes, an ambulance comes with the defibrillator prepared and transports the patient to the closest hospital. On the way to the hospital, the ambulance agent gathers the patient's symptoms and medical history from the patient's agent and informs the hospital reception agent about the patient's arrival. Having received the patient's symptoms and his medical history, the hospital reception agent looks up an available specialist, who is later helped by his own agent in checking and diagnosing the patient. From this point on, the specialist's agent monitors the patient by communicating with his personal agent."*

Unfortunately, it's a fact of life that people faint either in a park, or on a street or at home, without any possibility to alert a person or an emergency service. The scenario described above solves this issue, as the software agent is capable of sensing the environment (the patient's state of health) and autonomously alerting an emergency service when needed.

The authors of this paper have implemented a prototype agent with the following functions:

- simulates monitoring of patient's health state
- infers the action to be taken based on the patient's condition
- communicates with various prototype agents and web services in order to organize the emergency visit to the hospital

The prototype uses the Jade framework (jade.tilab.com) for developing software agents and the inference support in Jena (jena.sourceforge.net). Next steps should include employment of real ontologies for the agents and services to communicate semantically.

In its knowledge base, the patient's software agent has an ontological description of the symptoms and of other entities that are part of the medical domain and needed here. The file *patient.n3* shown below is a simplification of this description, used here for illustration purposes.

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>
@prefix : <http://www.example.com/#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix owl:  <http://www.w3.org/2002/07/owl#>

:Symptom a owl:Class .
:BloodPressure a owl:Class;
    rdfs:subClassOf :Symptom .
:Pulse a owl:Class;
    rdfs:subClassOf :Symptom .
:Service112 a owl:Class .
:val a owl:ObjectProperty .
:alert a owl:ObjectProperty .
```

Tab. 1. The simplified ontology – patient.n3 file excerpt

Basically this file defines BloodPressure and Pulse as a Symptom, while *val* and *alert* are properties of the symptoms and of the emergency service, respectively. Also, a list of RDF-based (w3.org) rules used by the Jena inference engine is part of the agent's knowledge base. An excerpt from the file *alert.rules* is shown below.

```
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix :     <http://www.example.com/#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix owl:  <http://www.w3.org/2002/07/owl#> .
```

```
[bradycardia: (:Pulse :val ?x), lessThan(?x, 60) -> (:Bradycardia :val "true")]
[tachycardia: (:Pulse :val ?x), greaterThan(?x, 100) -> (:Tachycardia :val "true")]
[call112: (:BloodPressure :val ?x), greaterThan(?x 140), (:Tachycardia :val "true") -> (:Service112 :alert "true")]
[call112_2: (:BloodPressure :val ?x), lessThan(?x 90), (:Bradycardia :val "true") -> (:Service112 :alert "true")]
```

Tab. 2. The RDF rules – alert.rules file excerpt

The first rule says *"if the pulse is less than 60, then the patient has bradycardia"* and the last rule *"if the blood pressure is less than 90 and the patient has bradycardia, then you have to alert the emergency service 112"*.

The agent permanently monitors the patient's condition, by receiving the parameters from the sensors connected to the patient's body. Each parameter is translated by the agent into a statement, e.g.:
:Pulse :val 110
:BloodPressure :val 145

Having these two statements and the rules above, plus the ontological description of the symptoms in Table 1, the agent infers the statement *Service112 :alert "true"*. This is how the agent autonomously determines that it should alert the emergency service when the patient's condition deteriorates.

Note that we use the N3 notation (w3.org) to describe the ontology. N3 has the advantage that it is human-readable, so any human operator can easily update it. This makes it easy to install and configure the agents on users' devices. Also, the N3 format can be used to describe the ontologies that are referenced by the web service descriptions in their semantic annotations. This is how semantic web can be integrated into the framework described in this paper.

In the example above (Table 1), we use a simple ontology for the sake of simplicity. In a real life scenario, a good candidate for a medical ontology is one based on HL7-RIM model (Orgun & Vu, 2005), (hl7.org). The great advantage is the worldwide adoption of this standard and thus the possibility to interoperate with medical institutions all over the world.

The example in this section highlights the capabilities the agents can leverage with minimum knowledge and little computing power. These very characteristics, together with the autonomy, make software agents suitable for the scenario described at the beginning of this section. Namely, software agents are typically used in embedded/mobile systems, having thus limited resources and intermittent Internet connection with little bandwidth.

The actions inferred by the agents can be much more complex than in this example. For instance, an agent may be interested in finding the diagnoses of all patients with similar symptoms as John (in the scenario above). For that, an agent needs a knowledge base and an inference engine.

3. MULTI-AGENTS AND SEMANTIC WEB

As suggested by the scenario in the previous section, software agents communicate seamlessly with other agents and with web services. Some services, like patient's medical record retrieval might be exposed online for authorized clients, while others, like negotiating an appointment, can be only done via agent communication.

The most important thing to notice in the scenario above is the use of ontologies. The patient's agent might be able to alert the emergency service, but what exactly does it communicate to this one? The intention is to communicate the patient's personal data (name, address, etc.), the symptoms, the medical history and the patient's location. Each of these aspects needs to be understood by parties working and communicating in a highly heterogeneous environment. The key is the usage of ontologies: the parties should either use the same ontology to express the concepts they want to communicate, or there should exist an ontology mapping between the ontologies used by the parties. An ontology mapping, as the name implies, would find concepts expressed in different ontologies either equivalent or 'similar enough', or different (i.e. there is no mapping between them).

Since the entities (software agents, web services) presented in our scenario are loosely coupled, the aim is to semantically discover each other. In other words, we intend to find an operation, given the semantic concepts of the inputs and outputs. As pointed out by various researchers (Christesen et. al., 2006), (Verma et. al., 2005), SAWSDL (w3.org) can be used to semantically annotate web services and the annotations can be recorded in UDDI V3 (uddi.org) using *tModels* and *categoryBags*. Semantic web services can thus still be registered in UDDI, just like standard web services. The discovery of web services in a UDDI registry consists of matchmaking the given profile (description of the functionality we are looking for) with the service descriptions found in the registry. Note that we are interested in a semantic discovery process, which means matching the input and respectively the output parameters of the services by meaning, rather than by syntax.

4. CONCLUSION AND FUTURE WORK

Building an automated patient monitoring system is possible with the current technologies. The key aspect is to combine the right technologies in order to obtain powerful frameworks that have the potential to improve the medical act. Namely, multiagent systems are interesting here because of their autonomy, mobility, limited usage of resources and capacity to communicate in a loosely coupled architecture. Their communication acts are based on shared ontologies or on ontologies between which a mapping exists. Semantic web can be employed to leverage legacy services and to be used by agents, as if they were just other agents. This seamless integration is possible because semantic publishing and discovering is conceptually the same for web services and agents. Future work includes solutions for issues such as privacy, security (the patient's medical record can only be seen by certain persons), performance, scalability (an ontology mapping service might become the bottleneck in such a system), reliability, availability and in general the non-functional aspects.

5. REFERENCES

Christensen, K.; Olesen, T. & Thomsen, L. (2006). Matching Semantically Described Web Services Using Ontologies, *124X Information Technology and Control, Vol.35, No.3A, ISSN 1392*

Dogac, A.; Laleci, G.; Kirbas, S.; Kabak, Y. & Sinir, S. (2005). Artemis: Deploying Semantically Enriched Web Services in the Healthcare Domain, *Information Systems Journal, Volume 31, Issues 4-5*, (June-July 2006), pp 321-339

Orgun, B.; Vu, J. (2005). HL7 ontology and mobile agents for interoperability in heterogeneous medical information systems. *Computers in Biology and Medicine 36*, 2006, pp 817–836

Verma, K.; Sivashanmugam, K.; Sheth, A.; Patil, A. & Oundhakar, S. (2005). METEOR-S WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services, *Journal of Information Technology and Management, Vol. 6, No. 1* (2005) pp. 17-39. Kluwer Academic Publishers.

*** http://www.hl7.org, The Health Level Seven Standard, *Accessed on: 2010-02-01*

*** http://jade.tilab.com/, Jade, Java Agent Development Framework, *Accessed on: 2010-06-03*

*** http://jena.sourceforge.net/, Jena Semantic Web Framework, *Accessed on: 2010-06-03*

*** http://www.w3.org/DesignIssues/Notation3.html, N3, *Accessed on: 2010-06-03*

*** http://www.uddi.org/pubs/uddi_v3.htm, UDDI

*** http://www.w3.org/TR/sawsdl-guide/, SAWSDL