Supporting Prenatal Care in the Public Healthcare System in a Newly Industrialized Country

Ingrid Nunes[1], Ricardo Choren[1,2], Camila Nunes[1], Bruno Fábri[1,3],
Fernando Silva[1,3], Gustavo Carvalho[1,3], Carlos J.P. de Lucena[1]
[1]Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Software Engineering Laboratory (LES)
Rio de Janeiro, Brazil
{ionunes,cnunes,lucena}@inf.puc-rio.br, {bfabri,fernd,guga}@les.inf.puc-rio.br
[2]Military Engineering Institute (IME)
Rio de Janeiro, Brazil
choren@ime.eb.br
[3]Prime UP – Quality in Software Engineering
Rio de Janeiro, Brazil

ABSTRACT

Most of women's deaths related to pregnancy occur in newly industrialized countries. In association with gynecologists and obstetricians of the Antônio Pedro University Hospital (HUAP) in Brazil, we have identified deficiencies in the prenatal care of the Brazilian public healthcare system that can be computer-supported. They are mainly related to protocols that must be followed in the primary healthcare institutions and the referral process that must take place when a high risk pregnancy is identified, besides other functionalities that can be automated by a software application. In this paper, the Prenatal Care Unified System (SUAP) project will be introduced, which provides a Multi-agent System for supporting and monitoring the prenatal care. This project uses agent technology to manage healthcare records, to act as a clinical decision support system, and to handle the logistics of high risk pregnancy cases. We also describe the challenges encountered during the implementation of the SUAP and discuss the benefits that an agent-based solution provided to the development of our system.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*Multiagent systems*; J.3 [**Computer Applications**]: Life and Medical Sciences—*Medical information systems*; D.2.11 [**Software Engineering**]: Software architecture—*Domain-specific architectures*

General Terms

Design.

Keywords

Multi-agent Systems, Prenatal Care, Healthcare, Applications, Software architecture.

Cite as: Supporting Prenatal Care in the Public Healthcare System in a Newly Industrialized Country, Ingrid Nunes, Ricardo Choren, Camila Nunes, Bruno Fábri, Fernando Silva, Gustavo Carvalho, Carlos Lucena, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1723-1730

1. INTRODUCTION

Around 600,000 women worldwide die each year from causes related to pregnancy and almost all these deaths occur in Newly Industrialized Countries (NICs) [14]. There is substantial evidence that healthcare during pregnancy is a crucial component of ensuring a safe delivery and a healthy mother and baby [8]. Indeed, women who receive prenatal care have lower rates of maternal and infant mortality.

The use of modern information and communication technologies to meet the needs of citizens, patients, and healthcare professionals (e-health systems [1]) can aid to decrease both the cost of prenatal healthcare services and also the load of medical practitioners. E-health systems collect patient data for later healthcare professionals' examination. In the case of regular prenatal care, a pregnant woman may present several conditions (disorders). For each condition, the professional should follow a given clinical practice guideline (prenatal care protocol), a process designed to assist professionals to make decisions about appropriate medical problems. These protocols aim to reduce inter-practice variations and cost of medical services, improve quality of care and standardize clinical procedures [5]. In this sense, e-health systems will be better exploited when they help the healthcare professional in the prenatal care decision process.

Multi-agent Systems (MASs) can be used to feed highly specialized healthcare professionals with the right information, at the right time, tailored to the patient [2]. In this paper, the Prenatal Care Unified System (SUAP) project will be introduced, which provides a MAS for supporting and monitoring the prenatal care. The term *Unified* is due to the unification of data in a central database, which may be accessed from different healthcare institutions. The project is developed by the Software Engineering Laboratory (LES) at PUC-Rio in association with gynecologists and obstetricians of the Antônio Pedro University Hospital (HUAP) in Brazil. The SUAP uses the agent technology to manage healthcare records, to act as a clinical decision support system, and to handle the logistics of high risk pregnancy cases. It provides functionalities with a pro-active behavior, which address issues related to the limited amount of resources available in the public healthcare system in NICs.

The main goals of the SUAP project include: (i) storing and accessing electronic healthcare records of the pregnant woman, so that the information can be used at any stage of prenatal care protocols; (ii) identifying and monitoring the application of prenatal care protocols, operating as an expert system that acts behind the scenes to help the healthcare professional. It is important to mention that, in some cases, a protocol should be adapted to be more effective. In such situations, the monitoring system can be used to show the effectiveness of a protocol to the professional, so that he can reason about changes; and (iii) supporting the contextualized access to prenatal care-related services. High risk pregnancy women should be referred to secondary healthcare institutions. This referral process depends on several aspects, such as the pregnant woman location, the clinical conditions of the mother and fetus, etc.

In the SUAP system, agents are responsible for monitoring the system and for acting proactively to provide these aforementioned functionalities. Besides describing the SUAP, we also discuss the challenges we faced during its implementation and the benefits of using an agent-based solution.

The remainder of this paper is organized as follows. In Section 2 we introduce our domain and highlight the main problems we are addressing with our system. Section 3 introduces and details the SUAP. Section 4 presents and discusses lessons learned with the development of our application. Finally, Section 5 concludes this paper.

2. PROBLEM STATEMENT

This section describes some of the problems that affect the prenatal care system in a NIC. These problems are specific to the Brazilian reality since they are related to how the prenatal care is structured in Brazil. The Brazilian prenatal care aims at providing a good assistance for pregnant Brazilian women, but it presents restricting resource limitations. In the following, we detail the main problems that the SUAP system currently addresses.

- **Pregnant women data is still mostly stored manually.** In Brazil, all prenatal care records are stored in a Governmental system (named "SisPreNatal"), which is responsible for prenatal appointments scheduling. The main purpose of this system is to know some statistical information in order to plan healthcare policies and budget distribution among cities and healthcare institutions. However, relevant data collected along the appointments, such as weight, cardiofetal beats and blood pressure, are registered in a paper "pregnant woman card" (*"Cartão da Gestante"*, in portuguese). Every pregnant woman has a card like this and she must carry it with her in all the appointments throughout the prenatal care. If the pregnant woman presents a condition that requires her to be transferred to a more specialized healthcare institution, she must also bring this card. From this scenario, it emerges the need for *electronic health record management for prenatal care systematization.*

- **Protocols established by the Government may not be appropriately followed.** During the prenatal care, several procedures must be performed either to prevent complications during the pregnancy or to react to some identified condition. In the public healthcare system, these procedures are established by the Government as prenatal care protocols. These protocols are derived from historical pregnancy cases and they are condition-specific. For instance, if a pregnant woman is diagnosed with syphilis, the healthcare professional should follow a protocol defined to treat syphilis. However, this protocol may be adapted over time. For example, the protocol to treat syphilis currently says the medical professional must prescribe 2.4 million units of penicillin. Previously, the protocols indicated a lower dosage of 1.2 million units, but it changed because it was concluded that this dosage was not appropriate, based on historical cases. Regarding to the protocols, there are two main issues: (i) they are supposed to always be followed. Therefore, to enforce the use of appropriated protocols, the *data collected during the prenatal care must be monitored and, when the context of a protocol is identified, there must be an alert requesting that the protocol activities must be performed*; and (ii) protocols may change. To allow this adaptation, *all prenatal cases must be constantly analyzed in order to verify when a protocol is not being sufficiently effective.* It is important to emphasize that adapted protocols can only become current practice just after Government approval.

- **It is not precisely known which is the most appropriate hospital to refer a high risk pregnant woman, based on her condition.** Pregnant women are not considered patients, because they are not sick. However, they must be monitored during the pregnancy in order to detect complications as soon as possible. This is important because certain characteristics or factors make a pregnancy high risk. Medical professionals identify these factors to determine the degree of risk for a particular woman and baby to provide better medical care. For instance, if a pregnant woman has high pressure, she must be correctly assisted to prevent eclampsia. If a pregnancy is low risk, the prenatal care does not need a more sophisticated infrastructure. This corresponds to the primary healthcare institutions in the Brazilian public healthcare system. Whenever a pregnancy is classified as high risk, the woman should be referred to a secondary health institution. However, there is the need to *identify which is the most appropriate secondary health institution according to the pregnant woman condition.* Figure 1 shows a canonical view of the referring problem.

Besides these problems, there are some other prenatal care activities that can be automated. These include: (i) *Ap-*

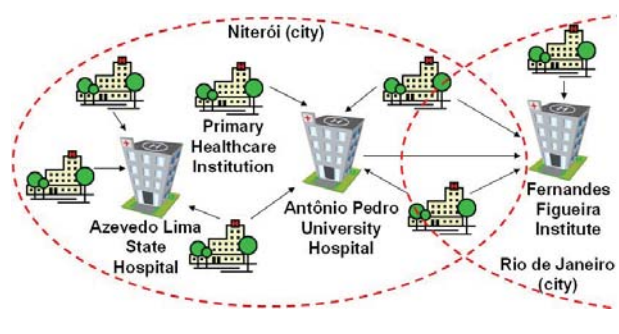

Figure 1: Canonical View of the Referring Problem.

pointment Scheduling Suggestion – appointments are scheduled according to a preset calendar, which can be changed in case a deviation (e.g. the weight of a pregnant woman is not inside the typical weight interval) is found. Based on this information, our solution is able to indicate when a pregnant woman should schedule her next appointment; (ii) *Medicine Incompatibility Alerts* – when a medicine is prescribed, the system should verify its side effects in the pregnancy and also other warnings such as if the pregnant woman has any allergy to medicine components; (iii) *Exam Reminder* – when a medical professional requests exams to a pregnant woman, the system reminds him to ask for the results in the next appointments; (iv) *Mandatory Notification Sender* – SUAP keeps a record of the diseases that must be notified to the Government. Whenever one of those diseases are diagnosed, the system sends a notification; and (v) *Notifications of Infectious Diseases* – the system should make proper notifications to the authorities whenever it detects possible outbreaks and epidemics.

We proposed the SUAP project to handle these problems. Our main stakeholders are the gynecologists and obstetricians of the HUAP, and they are the primary users of the system. In next section, we describe development details about the SUAP, and its architecture as well.

3. SUAP: Prenatal Care Unified System

The SUAP project consists of a web application integrated with software agents. The SUAP intends to show the technical feasibility of systematizing the Brazilian prenatal care as a Web-based application. Nonetheless, most of the SUAP requirements include pro-active behavior and, in some cases, reasoning and learning abilities. Therefore, we have chosen the agent technology to design and implement our system.

This section details the development of the SUAP. First, we present an overview of our proposed solution and go into the details of the SUAP design and implementation, presenting its architecture (Section 3.1), which is based on a previously proposed architectural pattern [11]. Next, Section 3.2 describes our development approach, which adopts some ideas of agile programming. In Section 3.3, we detail the agents that comprise the system. Finally, we present our first results in the SUAP development in Section 3.4.

3.1 SUAP Overview

The SUAP is mainly a Web-based system. However, some aspects of the system require autonomous and pro-active behavior, such as indicating situations for protocol usage and protocol monitoring. Thus, the SUAP architecture is twofold: (i) it has modules that represent the web application; and (ii) it is integrated with a MAS. This integration poses some challenges in the system development.

In order to address this issue, the SUAP has been developed following the Web-MAS [11] architectural pattern, which allows the construction of web-based systems with autonomous behavior provided by software agents. This pattern extends the Layered architectural pattern [6], which is a widely adopted pattern to structure web applications. The Web-MAS pattern provides a low-impact integration of software agents into a web application architecture.

Besides the web application, the Web-MAS pattern has three modules (detailed in Section 3.3): (i) *Business Layer Monitor*, which monitors the execution of business services and propagates them to the agents in the system. It represents the environment of the MAS; (ii) *Agents Layer Facade*, which is the access point for the Web application interact with the MAS, and; (iii) *Agents Layer*, which is composed of the agents of the system.

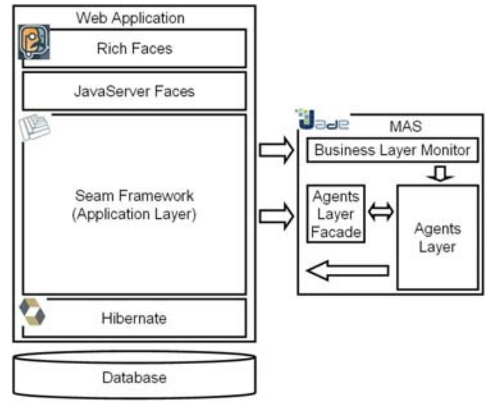

Figure 2: SUAP Architecture and Technologies.

During the SUAP development, we adopted a set of technologies that provide an appropriate infrastructure for the application. Figure 2 illustrates the structure of the SUAP architecture and technologies. We used the Seam[1], a framework for building rich Internet applications in Java, together with the JBoss application server. Software agents were developed with the JADE[2] framework. The single layer that we had to implement in the web application is the application layer due to the use of the Seam. It was not necessary to structure the system using the traditional three layers (presentation, business and data) because the Seam provides the encapsulation and implementation of the other layers.

The SUAP development team consists of ten members: 1 project manager, 3 analysts/designers, 1 software architect, 5 programmers and 1 web designer. The team members present several levels of experience since they include professionals and students (both undergradute and graduate).

3.2 Development Approach

The development of SUAP did not strictly follow any of the usual software development processes. However, we adopted some agile practices. Our development approach was built on the foundation of iterative development with continuous integration. Initially, we have developed functionalities that systematized the prenatal care workflow, which were implemented as a typical Web application. Then, we incrementally developed new functionalities using software agents. The first of these iterations required an additional effort to integrate agents into the Web system.

Our interaction with stakeholders was very intensive. This interaction included several meetings and interviews with the gynecologists and obstetricians from the HUAP, and also intense communication through email. Simplicity was present in the process as well, mainly in the documentation. We did not follow any particular agent-oriented methodology and we only documented what is really necessary i.e. we did not spend time and effort by making several models. Finally, we also employed the pair programming practice.

[1]http://seamframework.org
[2]http://jade.tilab.com

Figure 3 illustrates the process that has been guiding our development. The six steps are: (i) *Requirements elicitation*: identification of system requirements for the current iteration. This was accomplished mainly by meetings with the gynecologists and obstetricians; (ii) *Analysis*: analysis of the previously identified requirements. The following methods were used: use of examples, screen prototypes and informal agent models (discussed in Section 4.2). These techniques facilitated the communication with domain experts; (iii) *Planning*: definition of a general plan on how the requirements are going to be implemented, by making an overall analysis of the impact of the increment, the resources needed, and so on; (iv) *Engineering*: this phase consists of modeling the requirements. This was performed by either UML models or even by code skeletons; (v) *Construction & Release*: realization of the system modeling, ending in the release of a new version of the system, and finally; (vi) *Evaluation*: the prototype is validated by the medical professionals. First we make a presentation of the system, followed by their experimentation of the system. This step helps to correct problems in the prototype based on the professionals' feedback. Additionally, it is important to collect new requirements with the medical professionals and to start a new iteration. This process has been proven very effective, due to our poor initial knowledge about the prenatal care domain.

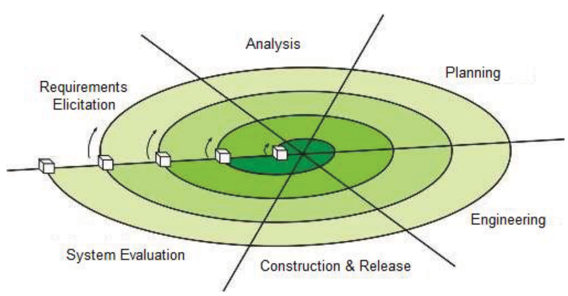

Figure 3: SUAP Development Approach.

3.3 Detailing the SUAP Agents

Section 3.1 presented a big picture of the SUAP architecture. In this section, we detail the MAS module of the architecture (Figure 4). This module has two submodules that connect the web application to the MAS: *Business Layer Monitor* and *Agents Layer Facade* submodules. The submodule that is in charge of providing the autonomous and pro-active functionalities of the SUAP is the *Agents Layer*.

The *Business Layer Monitor* and *Agents Layer Facade* are implemented by the `Environment` and `Facade` agents, respectively, which are responsible for bridging the gap between the web application and the MAS. The connection between the `Environment` and the web application is achieved through the use of the Observer pattern [7], which enables the loose coupling of the Web application and the MAS. The `Environment` agent is an observer of the business execution processes, which are observable entities. This is the way agents perceive the environment. On the other hand, agents change the environment by accessing the *Application Layer* to perform changes in the database. The `Facade` agent, in turn, is used by the web application to request information from agents in the MAS, thus hiding implementation details of the *Agents Layer*.

The *Agents Layer* module consists of two parts: the *Core* and the *Protocols*. The *Core* agents provide the base functionalities of the SUAP, e.g. suggesting the date of the next appointment of the prenatal care or prescribing a medicine. They provide atomic services – atomic in the sense that they do not need other agents to provide a service or achieve a goal. The *Protocols* agents, as the name suggests, are responsible for monitoring situations of the prenatal care in which protocols must be applied and for taking appropriate actions when such situation is detected. The actions are typically related to requests that will be made to the *Core* agents. Next we detail each of the *Core* agents:

Scheduler. The prenatal care in the Brazilian public healthcare system has a preset minimum schedule for the low risk pregnancy. However, according to some situations (expressed in protocols) this schedule can be changed. The `Scheduler` is responsible for suggesting the next appointment at the moment that each appointment finishes. This agent receives requests from other agents to change the preset schedule.

Pharmacist. The `Pharmacist` agent has two main goals: to suggest medicine prescriptions and to verify possible drug allergies and interactions that can be harmful. When an agent wants to suggest the prescription of a medicine, it must send a request to the `Pharmacist`. Incompatibilities are verified when either a medical professional or an agent prescribe a medicine.

Exams Manager. This agent suggests exams that need to be requested according to the gestational age or the results of previous exams. Exams may also be requested due to protocols. In this situation, *Protocols* agents verify such situation and send a request to the `Exams Manager`. This agent is also responsible for reminding the medical professional about pending exam results.

Analysis and Statistics. This agent provide reports about the prenatal care. Most of this information is needed by the Government. In addition, the `Analysis and Statistics` agent receives a notification when a protocol is executed, so it can monitor if the problem that the protocol is addressing is solved, i.e. it monitors the effectiveness of protocols and it makes reports on that.

Health Official. There is a list of diseases that must be informed to the Government as soon as they are detected or confirmed. The `Health Official` agent detects when a disease is diagnosed (this is always performed or confirmed by a medical professional), and it creates a report that must be sent to the Government.

Logistics. When *Protocols* agents detect that a pregnant woman must be referred to a secondary healthcare institution, it sends this information to the `Logistics` agent, which indicates an appropriate institution. Currently, it makes suggestions based on a set of rules.

The *Protocols* agents monitor the data that is being informed in appointments and exam results to identify situations in which protocols must be applied. Protocols are defined by a set of rules, e.g. the Weight Protocol has rules to verify if the pregnant woman's weight is either higher or lower than the ideal. Rules related to the five main

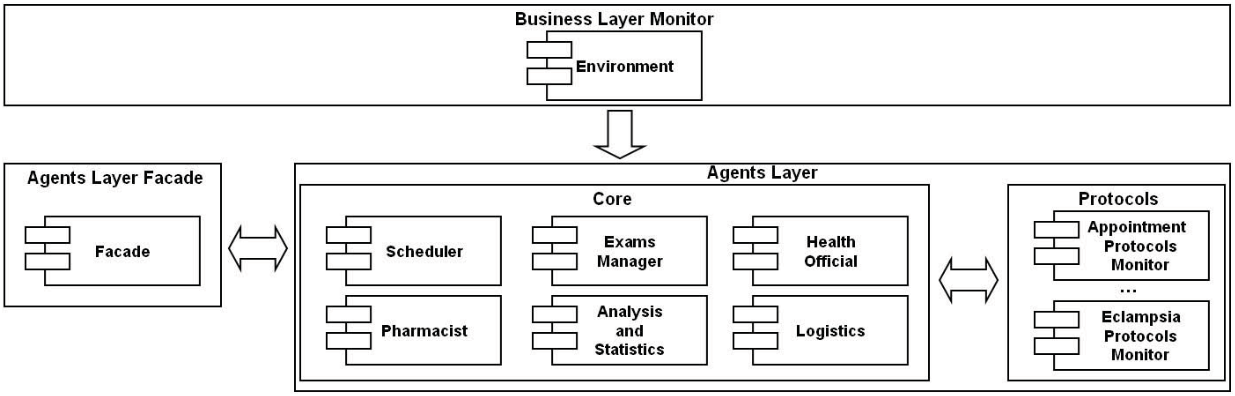

Figure 4: SUAP Agents.

measures performed in an appointment (weight, cardiofetal beats, uterine height, pressure and edema) were represented in a similar structure (`<protocol, event, context, behavior>`), which are part of the `Appointment Protocols Monitor` knowledge. When this agent perceives an `event` from the environment, it checks if the `context` matches (gestational age, value of the target measure, and so on), and if so, it executes the `behavior`. This behavior consists of the following actions (or a set of them): (i) perform message alerts; (ii) suggest the prescription of a medicine; (iii) suggest an exame; (iv) change the preset schedule; and (v) refer the pregnant woman to a secondary health institution. All the actions but (i) result in messages sent to *Core* agents. These rules were carefully analyzed with gynecologists and obstetricians. Rules are minimalist, in the sense that they are part of the system only if they must be executed in all cases. The other protocols, such as Eclampsia, Gestational Diabetics, Syphilis and Toxoplasmosis, do not follow this same structure. They are currently being analyzed with gynecologists and obstetricians to be developed, and are going to be incrementally implemented in specific agents.

It is important to notice that no agent takes a decision on behalf of a medical professional. An agent can only make suggestions (e.g. prescribe a medicine, request an exam), helping the professional to make decisions. This is a requirement of our stakeholders.

3.4 Results

The SUAP is currently deployed into a staging environment, which is used to assemble, test and review new versions of a website before it goes into production. It is being used experimentally by gynecologists and obstetricians from the HUAP. Our system is being incrementally developed (Section 3.2), thus each new functionality developed is integrated into the system and then a new version is deployed in the staging environment. The first version was released in July 2009 and it was composed of the core functionalities that provided the the prenatal care systematization.

Currently, all agents that are not part of the `Protocols` module were developed. From this module, only the `Appointment Protocols Monitor` agent has been already implemented. In addition, as soon as the system collects and stores large amounts of real data, we are going to start experimenting with machine learning algorithms to improve the suggestions in the referral process. With a large data set, it is possible to make cross-validation to verify and compare the quality of the results by adopting different models.

Figure 5 shows a screenshot of the SUAP, which presents the step of registering a typical pregnancy appointment. It illustrates the situation in which the weight is not adequate to the current gestational age. Therefore, it can be seen in Figure 5 an alert indicating this nonconformity.[3]

4. DISCUSSIONS

The SUAP development provided us the experience of building a MAS from scratch, with the challenge of identifying the scope and requirements of a domain previously unknown to the developers. For this task, there was the need of intensively interacting with domain experts (gynecologists and obstetricians). This section discusses the challenges (Section 4.1) and benefits (Section 4.2) of using an agent-based solution in the development of SUAP, and present related work (Section 4.3) as well.

4.1 Challenges in MAS Development

In our laboratory, we have been developing agent-based systems for the past few years. In such software, our main goals are concerned with the evaluation of particular approaches of agent-oriented research. The domain of these case studies are the typical ones used in the MASs literature, such as conference management systems and e-commerce. In addition, the agent-like functionalities of these systems, i.e. the ones with pro-active and autonomous behavior and/or requiring reasoning and learning capabilities, are well-defined.

A main difference between these case studies and the SUAP is that we did not have a previous knowledge about the domain and we had little idea what we could do to improve the prenatal care. Moreover, the domain experts did not know what they need as well – they did not know what the system could do for them, besides providing the informatization of what was being done manually. We believe this situation happened because our stakeholders are not familiar with systems that automate their tasks nor are able to reason about data. Therefore, the first main challenge we faced was to *discover how we can improve the prenatal care, i.e. which tasks can be automated and delegated to our system.*

[3]Alert in English: Subnutrition. Investigate food history, hyperemesis gravidarum, infections, parasites, anemia, debilitating diseases.

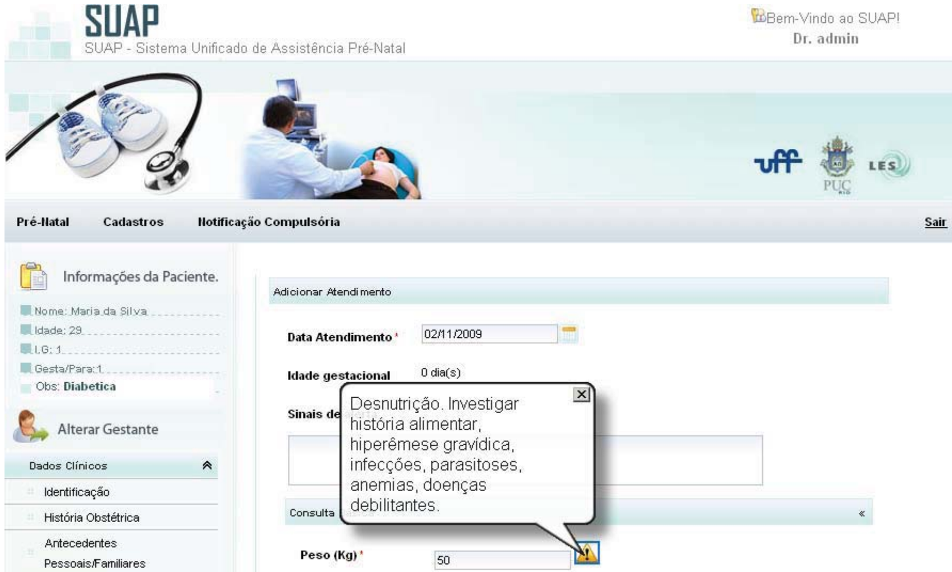

Figure 5: SUAP Screenshot – Alert of Low Weight During an Appointment.

Another problem we dealt with was the scope identification of our automating tasks. Having agents acting on behalf of users may bring the idea that *automating everything that is possible is good*. We have concluded that *this is not true*, based on discussions with our stakeholders. During the elicitation of the SUAP requirements, medical professionals showed high interest in most of our suggestions, and we inferred they also wanted a system that uses case-based reasoning to suggest diagnostics. This functionally is very common in expert systems for the healthcare domain. Nevertheless, the medical professionals do not want this functionality in the SUAP. When we are dealing with the public healthcare system, medical professionals must strictly follow protocols and the system should identify when a protocol must be applied and indicate which action should be performed. No protocol can be changed automatically by the system, even though it is not being regarded as effective.

These two issues are related to requirements elicitation focusing on identifying functionalities that automate the prenatal care, or require reasoning and learning techniques. In addition, it was challenging to *find a solution on how this autonomous behavior, i.e. agents, should interact with users.* In our system, agents must express notifications about protocols, medicines, exams and so on, but at the same time they must not be annoying. Some medical professionals are experts thus they do not need the system to give notifications. In SUAP all notifications are given by displaying an alert icon on the screen (see Figure 5), and the corresponding message is only displayed if the user points to this icon. Some of the alerts require the professional to provide a reason if the suggested action is not followed. However, the system does not preclude him of doing any action.

From an engineering point of view, the major challenge that we faced was modeling the domain as well as understanding and representing protocols (knowledge engineering). The information that we have about actions to be performed for each concern (weight, uterine height, and so on) and the context of their application are very specific, and it was difficult for us to find generic concepts to be used in the protocol representation. We aimed at modeling protocols in a generic way in order to reduce the impact of protocol creation and update (when the Government changes a protocol). On the other hand, the content of the two main manuals that we have to follow [4, 9] helped us due to the structure they present the information about protocols.

As we adopted an incremental development process (Section 3.2), a good modularization of the architecture and its modules is essential to allow a smooth system evolution and maintainability. This was particularly challenging because most MAS applications, even those based on widely-known frameworks, are not concerned with stability and maintainability principles during the conception and implementation of MAS architectures [10]. Therefore, our goal is to not only take advantage of the MAS abstractions to model autonomous and pro-active behavior, but also to take into account software engineering principles in order to allow the development and design of a stable and maintainable MAS architecture, thus facilitating the system evolution.

Regarding the SUAP implementation, a main issue was choosing the agent platform. First, we considered using a belief-desire-intention (BDI) platform for providing a reasoning engine to our agents. However, after identifying our requirements, we realized that our agents provide a proactive behavior based on a set of fixed rules, and *the BDI architecture does not add advantages to them*. In addition, we were concerned with our programmers background. Even though they have large experience with object-oriented technologies, they did not have previous knowledge about MAS and implementation platforms. As a consequence, *adopting complex platforms that provide several new abstractions would significantly increase training costs.*

Based on these facts, we have chosen JADE to develop our agents. JADE is basically based on two main abstractions (`Agent` and `Behavior`), which were sufficient for our problem. Furthermore, agents are coded in "pure" Java (i.e. not in XML, like Jadex), and this reduced the learning curve of our programmers. Members from our team who have been working with MAS in the past few years trained the programmers. Less experienced programmers also studied the JADE platform mainly with the documentation provided by it. The programmers reported that it was not hard to learn and understand JADE.

Another essential constraint that we considered on choosing our agent platform is the *integration with object-oriented technologies*. Several successful object-oriented frameworks have been developed for web application development, which significantly reduce development time. In our system, Seam was adopted because it reduces the complexity of designing architectures, by providing benefits such as simplicity in the applications development, which allows reducing the number of lines of code and the use of code annotations to provide an easy transition of objects among the framework layers. Using a Java-based agent platform, such as JADE, facilitates the integration with object-oriented frameworks.

Nevertheless, the integration of JADE and Seam frameworks was not trivial. One of the functionalities provided by the Seam is persistence, which has to be used by agents. In order to access the database by means of the Seam, agents must be Seam components and be instantiated under the Seam container. However, JADE agents did not work as Seam components, and we believe this is due to thread control. We did not make further investigations on this, because we have adopted an alternative solution: we created a Seam component as a singleton instance used as a locator for components that can access the database for agents.

4.2 Benefits of Using an Agent-Based Solution

With the development of the SUAP we were able to identify the benefits of using an agent-based solution not only from the developer perspective, but also from the stakeholders point of view. A major benefit that we have identified is the *communication facility between analysts/developers and stakeholders,* mainly domain experts. As discussed in Section 4.1, our previous knowledge about the domain was very limited in the beginning and, besides the simple systematization of the prenatal care process, there was little idea on how we could improve the process. The agent-oriented abstractions showed to be a very useful to understand and model the SUAP from an analysis point of view, because they are similar to the "real world" abstractions and our domain experts were able to understand them.

We first sketched some ideas in very informal models, based on information we read in [4] and [9] (prenatal care manuals from the Government). Two models were used: (i) *Role model* – roles represented by boxes and inside them there are goals described in natural language sentences, which are the roles' responsibilities. Lines linking two roles represent collaboration between them; and (ii) *Agent model* – agents represented as packages stereotyped with ≪*agent*≫ (from UML) and the previous identified roles are represented as use cases stereotyped with ≪*role*≫ inside agent packages. It means that an agent plays the roles that are inside its package. A role, in our system, is used to modularize a certain responsibility. When the role is attributed to an agent, it means that this agent must have a behavior to accomplish the goals of this particular role.

Next, we introduced the agent concept to gynecologists and obstetricians, defining it as a pro-active, autonomous and situated entity with communication abilities. We made an analogy of agents and roles with the real world. Then we showed them our informal models. They understood the models quite quickly and they made valuable and useful suggestions thus helping us to model the domain by identifying requirements in terms of goals.

Moreover, in Agent-oriented Software Engineering, agent abstractions may be used not only in the analysis phase but also in the design and implementation phases. Providing that we use the same abstractions in these different phases, *the gap between analysis and implementation is reduced*. In the SUAP, we analyzed the domain in terms of roles and agents, and just refined them to be implemented. We have specified how they achieve their goals (plans) and the necessary knowledge to execute these plans (beliefs). Besides an agent-oriented engineering approach, other topics from MAS research are helping us in the development of the SUAP, such as machine learning techniques. In addition, we used existing frameworks and libraries to develop pro-active and autonomous behavior in the SUAP.

The adoption of the Web-MAS architectural pattern provided a structure to build the SUAP in a stepwise fashion, in which agents were incorporated in the system when the web-application component was already developed. Additionally, the design choice of different agents (loosely coupled components) for each protocol allowed the incremental development of the system as well. These evolutions in the SUAP were performed with a low impact, indicating the stability of its architecture.

Furthermore, to implement the SUAP architecture, we have used several existing technologies that provide modular support to the construction of maintainable MAS. All the used infra-structure – web and MAS platforms (Section 3.1) – are essential to guarantee the modularity of the SUAP implementation. For example, JADE and JBoss platforms already provide an adequate thread of control to manage agents and Enterprise Java Beans (EJBs) components, respectively. These platforms also offer alternative mechanisms to implement and modularize functionalities such as communication among agents in JADE and security in JBoss. In addition, JBoss also provides a load balancing feature, which helps on the system scalability. We have not made additional effort on this issue, because the SUAP has a low number of agents. Therefore the system bottleneck would be the number of web requests, what has been addressed by web and application servers.

4.3 Related Work

The idea of systematizing healthcare protocols is not new. In the past two decades, the EON[4] project has been addressing this issue. EON is an extensible architecture for developing decision-support tools for various aspects of protocol-based care. However, as discussed in [12], transforming narrative guidelines into a computer-interpretable formalism is still a bottleneck in the development of decision-support systems. Even though the EON project has made valuable contributions on knowledge representation[13], our major challenge was still understanding these protocols to model them. Patterns [12] may help to model protocols, but we did not find any that is useful for our domain.

A multi-agent model is presented in [3] for managing patients in the breast cancer domain, using as an example, a model of a breast cancer referral system. The idea is to refer patients to clinics that provide specific services needed. In addition, each clinic is independent, managing its own data.

[4] http://bmir.stanford.edu/projects/view.php/eon

In our case, the SUAP has a unified database system and during the referral process it chooses one of the secondary healthcare institutions that is better for the woman. It is not expected that this referring route stabilizes over time, as in [3], because in practice some healthcare institutions deal better with specific kinds of pregnancy complications.

One difference from these works to ours is the application domain. Dealing with pregnancy is different because a pregnant women is not a patient that has a disease to be treated, but a healthy person that must be monitored to avoid possible (fatal) complications, i.e. the goal is to detect potential scenarios of problems and not to solve them.

5. CONCLUSIONS

Several issues are being currently discussed by worldwide organizations such as global warming, economic crisis and healthcare system. In the particular case of NICs, healthcare presents some challenges due to resources limitation. Software systems can be used to facilitate information processing and to support medical professionals to execute tasks regarding clinical protocols.

This paper presented the SUAP, a MAS to support and monitor prenatal care. SUAP manages electronic healthcare records of pregnant women; models, monitors and provides advices on prenatal protocols; and, models a simple referencing protocol based on pregnancy risks. The main ideas were to increase the accessibility to healthcare records, help professionals to make right decisions and contextualize access to healthcare services. It is not a goal to make clinical decisions - these are left exclusively to the professionals.

All these aspects are particularly important in a NIC scenario, such as in Brazil. An electronic healthcare record database can help identify regional-specific diseases and conditions so that healthcare policies can be devised properly. Monitoring protocols is also important to improve the assistance based on historical data. Moreover protocols are created according to the available resources, and helping medical professionals to follow protocols eases the burden of the healthcare system as a whole. Referencing is also important as the pregnant woman shall be better assisted. To provide protocol and referencing functionalities, SUAP uses reasoning and learning capabilities of agents.

Another aspect of SUAP is that it is a Web-based system. In this sense, the SUAP development faced the challenge of providing a seamless integration of software agents into a Web-based application. Thus the SUAP architecture has two modules: (i) a MAS module, which is in charge of protocol monitoring, appointment scheduling, referral process, and so on, and; (ii) a Web-based module deals with the user interface, providing a simple access point to the system. These modules were integrated using two layers (*Business Layer Monitor* and *Agents Layer Facade*), as proposed in the Web-MAS pattern, that needed to be developed.

We have described our experiences in applying agent-oriented concepts to the engineering of a MAS for prenatal care. We followed an iterative process. After the elicitation phase conducted with gynecologists and obstetricians of the HUAP a prototype was built up using existing frameworks such as JADE and Seam. The results gained during the development of SUAP were quite promising. However we also found some challenges when applying agents to the development of real healthcare systems. After this experimental period of the SUAP in a staging environment, the first goal is to deploy it in a production environment and be used in the HUAP. The next step is to adopt it in Niterói city and then, hopefully, in the state of Rio de Janeiro. In this case, it is extremely important to submit the system to stress tests and verify concurrency issues.

6. ACKNOWLEDGMENTS

This work is a result of support provided by the AGENTESGRA project (Ref. 0714/07) funded by the FINEP. The authors would like to acknowledge the contributions of their colleagues: Bruno Dias, Filipe Daflon, Juliana Imperial and Vitor Pinheiro. We would like to thank our domain experts from the HUAP: Renato Sá (coord.), Cristina Robichez, Erica Ciasca, Thaline Valory and Valéria Moraes. We are also thankful to Maarten Sierhuis for his valuable feedback in preparing the final version of this paper.

7. REFERENCES

[1] European eHealth Ministerial Declaration, May 2003. Available at http:/ec.europa.eu/information_society/eeurope/ehealth/conference/2003.

[2] F. Bergenti and A. Poggi. Multi-agent systems for e-health: Recent projects and initiatives. In *WOA'09*, 2009.

[3] E. Black. Using agent technology to model complex medical organisations. *AgentLink News*, (12):7–8, 2003.

[4] M. L. Fajardo and et al. *Assistência pré-natal: normas e manuais técnicos*. Ministério da Saúde, Brasília, Brazil, 3rd edition, 1998.

[5] D. M. J. Field and K. N. Lohr. *Guidelines for Clinical Practice: From development to use*. Institute of Medicine, National Academy Press, 1992.

[6] M. Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley, November 2002.

[7] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley, 1995.

[8] W. H. Guilford, K. E. Downs, and T. J. Royce. Knowledge of prenatal health care among Costa Rican and Panamanian women. *Revista Panamericana de Salud Pública*, 23:369 – 376, 06 2008.

[9] Minas Gerais. *Atenção ao pré-natal, parto e puerpério: protocolo Viva Vida*. Secretaria de Estado de Saúde, Belo Horizonte, Brazil, 2006.

[10] I. Nunes, E. Cirilo, D. Cowan, and C. Lucena. Fine-grained variability in the development of families of software agents. In *EUMAS'09*, Cyprus, 2009.

[11] I. Nunes, U. Kulesza, C. Nunes, E. Cirilo, and C. Lucena. Extending web-based applications to incorporate autonomous behavior. In *WebMedia 2008*, pages 115–122, Vila Velha, Brazil, October 2008.

[12] M. Peleg and S. W. Tu. Design patterns for clinical guidelines. *Artif. Intell. Med.*, 47(1):1–24, 2009.

[13] S. W. Tu and M. A. Musen. Modeling data and knowledge in the eon guideline architecture. In *MedInfo 2001*, UK, 2001.

[14] World Health Organization and United Nations Children's Fund. Revised 1990 estimates of maternal mortality: a new approach by WHO and UNICEF. Document WHO/FRH/MSM/96.11; UNICEF/PLN/96.1, WHO, Geneva, April 1996.