On the Design of a Management Platform for Antibiotic Guidelines in the Intensive Care Unit

Femke De Backere, Kristof Steurbaut, Filip De Turck
Ghent University - IBBT,
Department of Information Technology (INTEC)
Gaston Crommenlaan 8, bus 201, 9050 Ghent, Belgium
Email: femke.debackere@intec.ugent.be

Kirsten Colpaert, Johan Decruyenaere
Ghent University Hospital,
Department of Intensive Care
De Pintelaan 185, 9000 Ghent, Belgium
Email: kirsten.colpaert@ugent.be

Abstract—Clinical guidelines are used in the Intensive Care Unit to assist physicians and nurses in taking diagnostic or treatment decisions. Although these guidelines can be transformed into a computer executable format, they often are handwritten and not in a standardized format, which makes it difficult to convert them into working services. Moreover, manually translating guidelines can cause communication problems between software developers and the medical staff. Problems can also arise in the integration of clinical decision support into the clinical workflow and the uptake by doctors. To counter this, a modular, distributed, multi-tier framework was developed for translating guidelines into software applications and providing clinical decision support in the Intensive Care Unit. Different requirements were taken into account. The architecture has been implemented using Java Enterprise Edition. A service-oriented approach is used, allowing an easy introduction of new functionalities and integration with other systems. The architecture was evaluated with the antibiotic dosage guideline, which is used on a daily basis in the Intensive Care Unit.

Keywords—Guideline management system; clinical guidelines; Java EE; UML; antibiotic dosage

I. INTRODUCTION

Due to the increase of sheer size as well as complexity of medical information and data, the need for guidelines to structure best practices becomes more important, since medical therapies keep improving and augmenting. At the same time, doctors and nurses are responsible for a larger number of patients than ever before. In clinical practice, variance in treatment should be avoided. Evidence-based practices should be applied to guarantee a high quality of care, while minimizing hospital costs. Using clinical guidelines can provide a solution to these problems. A clinical guideline [1] is a systematically developed document aiming to structure a treatment or diagnostic process. It contains conclusions and recommendations. Hospital or department specific details are deliberately left out, because in specific settings the guidelines can be deviated from, based on hospital policy or expert knowledge. Guidelines are always evidence-based and can be very extensive, but they should not be considered as rigid or strict directives.

The adherence to clinical guidelines has shown positive effects on practitioners performance [2]. The rate of patient referral is higher, the time to transfer to the appropriate unit or the time to diagnose a patient decreases. Nowadays, guidelines are often handwritten, text-based or structured as flow charts. When these guidelines consist of too many details, problems may arise regarding the size of the guideline. But, when the guideline is described rather superficially, it can become too vague to interpret. This can lead to the non-compliance of medical staff to guidelines. Adopting a common format for representing computer-interpretable guidelines (CIG) can provide a solution for these problems. Using CIGs, the guideline can be automated to a maximum extent, both in terms of description and execution, thereby providing an advanced level of clinical decision support.

The adoption of CIGs remains low, because of the communication gap between the medical domain and ICT experts [3]. Moreover, developing guidelines is very time-consuming and requires domain-specific knowledge from software developers and physicians [2]. One of the main reasons for low adoption is that guidelines often contain ambiguities and simplifications. They don’t cover all possible situations and are thus incomplete. Another problem for not computerizing guidelines is the existence of multiple formats for representing guidelines, each with their own advantages, specifications and goals. It is difficult to decide, which format is the most appropriate in a given situation.

A guideline management system can overcome the problems described in the previous paragraph, by offering a coordinating framework, able to translate, interpret and transform guidelines in working reusable services, handling the follow-up of both patients and guidelines, communications and calculations. Translating guidelines into working applications, providing clinical decision support, will benefit patients, reduce hospital admittance and therefore minimize costs. Physicians will be able to react quicker to changes into patient’s parameters and save time [4]. A service-oriented approach can be used to delimit the responsibilities of a particular service, as different guidelines tend to employ similar functionalities [5]. The chosen strategy can be summarized as follows: “Tight cohesion, loose coupling”.

By using a common format for representing clinical guide-
lines the gap between technical and medical domains can be minimized [6]. In this framework, UML (Unified Modeling Language) is selected as the modeling language [7], since it is well-known by software engineers and has similar model structures as the flow charts, currently used in hospitals. Besides, this language offers numerous other advantages. UML is very straightforward, in a sense that non-technical specialists have little difficulties with understanding UML diagrams and designs. Using this notation as a basis for decision support systems in the Intensive Care Unit (ICU) will reduce the non-compliance and optimize the treatment of critical ill patients. Implementing new guidelines will be less time-consuming and error-prone:

- All important aspects of a system under investigation can be described, thanks to its comprehensiveness.
- As a well-adopted standard, it ensures the interoperability and transformability of crafted guidelines.
- UML offers several diagram formats, enabling to express both structural and behavioral aspects of guidelines, similar to the traditional flow chart structures.
- If one studies the language in sufficient depth, formal designs can be created, from which executable program code can be generated.

The computerization of guidelines has several advantages. The most significant benefit is the integration with existing systems, such as the electronic health record (EHR) of the patient and hospital databases. Reminders or notifications can be used to inform physicians about the patients’ actual condition, which contributes to their follow-up. The main difficulty remains in decomposing a UML-based guideline into an orderly execution of basic services [6]. But an elaborate discussion of this topic goes beyond the scope of this paper.

This guideline management system allows physicians to provide clinical guidelines in a computerizable format, namely UML, which can automatically be transformed into a Java-executable guideline. The flow chart of the antibiotic dose guideline [8] was used to test this new system. It is expected that this system will also accelerate the development process of other medical decision support guidelines, by using UML as a common structured format for physicians and developers.

The management of antibiotics in the ICU consists of important guidelines. These guidelines offer a dosage advice, limit antibiotic duration (to avoid resistance) or give advice to switch to other antibiotics. The antibiotic dosage guideline calculates every day, which type of antibiotic has to be administered, as well as the corresponding quantity. Specific parameters will influence the dosage and type of a certain antibiotic given to a particular patient. The results will be shown on the bedside terminals of the patient. When a physician doesn’t respond within 2 hours to the recommendations, a reminder will be sent to his/her DECT phone, asking to reply on given suggestions.

The remainder of the article is structured as follows. In Section II the guideline translation methodology is concisely presented. This constitutes step one into our guideline lifecycle: from custom format to UML notation. Section III is devoted to the requirements of the decomposition and execution framework, its architecture and related implementation details. The UML guideline diagrams are treated as input, to end up with a running decision support system. To demonstrate a practical use case, the antibiotic dosage service is demonstrated in Section V. Evaluation details will be presented in Section VI. Finally, in Section VII the main conclusion of this research is highlighted and possible directions for future research are described.

II. GUIDELINE TRANSLATION METHODOLOGY

As we already outlined in the previous section, we opted for UML as a common representation format for guidelines that have to be automated. In this section, it is argued why this choice was made.

Examples of already existing formats for clinical guidelines are numerous. Some examples are the Arden Syntax, Gello, Asbru, EON, GLIF, GUIDE, PRODIGY, and PROforma [9] [10]. However, these formats are designed with the focus on the specific goals and requirements of the development and research team. This makes it more difficult to use them to represent different guidelines used in the ICU. Stated differently, each format has its strengths and weaknesses. Moreover, they often can only represent one type of guidelines [9]. Currently only the Arden Syntax and Gello are standardized by Health Level 7 (HL7). These formats offer assistance for clinical decision support in the form of a expression language and rule language. Gello is based on the Object Constraint Language (OCL), a constraint and query language for UML class models [11]. However, from the point of view of the physician, it is preferable to select a language with intuitive diagrams, which requires a very short period of time for physicians to show the ropes.

Being a standardized general-purpose modeling language, UML offers different types of diagrams. Activity diagrams can be used to depict the workflow of the guideline. They can describe the state of the activities by showing the sequence of the activities performed, conditions and loops can be added and tasks can be performed in parallel. Besides activity diagrams, interactions and state diagrams could also be used as both of them incorporate notions of order and time. As an added value, inconsistencies and shortcomings become more explicit in UML.
Figure 1 shows an extract of a flow chart used in a guideline and the corresponding UML representation. Few differences can be found between the two formats. The main differences are:

- The shapes in the flow chart representation are inconsistent. A decision node can be represented by a rectangle, whereby the arrows contain the possibilities. But this node is also represented as a triangle or a rectangle, containing the possibility.
- The actions are also depicted in different shapes.
- Not all the possibilities are covered. The flow chart doesn’t cover the situation when the time is exactly 24 hours. This can seem as a small problem. However, when translating this flow chart into a working program, the possibility will be left out. This can cause errors or exceptions during execution. Moreover, which condition tells us how we must alternate between more than 24 hours, or less than 24 hours?

These inconsistencies and simplifications can lead to difficulties when translating them to a working program. Using UML, these problems can be solved by eliminating the simplifications, ambiguities and errors in the guidelines.

III. REQUIREMENTS ANALYSIS

Some critical features should be embodied in a system for providing decision support, increasing the uptake of physicians [12] [13]. A first functional demand is the integration of automatic decision support into the workflow of the staff. Information originating from measurements, such as body temperature, heart rate and blood pressure, is processed by the platform and leads to appropriate messages destined for care providers. Secondly, the framework should not only observe someone’s current medical condition, but also supply various recommendations, with a corresponding motivation.

Next to these functional demands, quality attributes are used to describe the requirements of the platform itself:

- **Performance** can be addressed by prioritizing requests. Medical message notifications should prevail over uploading a guideline.
- A user friendly interface should be offered to the medical staff, this will improve the usability of the platform. A test environment can be developed to enable testing of the translated guidelines.

- By using a service-oriented approach to delineate a service’s responsibility, we ensure our application is open for future modifications.

The guideline management system is able to perform three tasks:

A. **Guideline adoption**

The first task of the guideline management system is the decomposition of the guideline into services. The guideline adapting system will interpret the UML activity diagram. This component will automatically decompose the clinical guideline into blocks. These services can already be in place, or will be created as the need arises. They should be considered as building blocks, responsible for a specific task such as measuring a specific parameter or registering a treatment action performed by a nurse or doctor. The data lookup service is responsible for the communication with the external database. It will query the database and return the results.

B. **Guideline management**

The system’s second task is the composition of the services into a working application, with the correct flow. Some services have to be triggered at a specific moment, which is achieved through a timer or scheduler component. The notifier service will alert a physician, by means of sending emails or text messages to the appropriate persons. In the ICU of the University Hospital in Ghent, each bed is equipped with a computer, called the bedside terminal. On this computer, informative applications and device output concerning the actual status and required actions can be visualized for a specific patient. For that reason, we need a web view.

C. **Reporting**

The ICU is dealing with a very large amount of data, coming from lab results, monitors, prescriptions and observations. Synchronization and interaction with external data is very important. The data, required by the system, has to be synchronized and preferably stored in a new database. In this way, it is easier to log specific data and export results as a CSV or Excel file, which will enable data mining to facilitate clinical studies and future improvements.
IV. ARCHITECTURE DESCRIPTION

Java EE 5 (Java Enterprise Edition 5) can be used to develop a service-oriented guideline management system. This platform defines a standard for developing multi-tier applications, based on standardized, modular components. A complete set of services is provided to these components and details governing middleware activities are handled automatically, without complex programming [14].

The platform has a multi-tier, distributed application model. Based on the function, the application is divided into several components. These components can be installed on different machines, depending on the tier they belong to. Generally, the enterprise application is split into a 3-tier or 4-tier architecture [15]:

- The client tier: the components run on a client machine. This can be the visualization of dynamic HTML pages or a application client.
- The web tier consists of Java Server Pages (JSP) and servlets. They provide a fast way to represent dynamic web content and enforce a separation of the representation of the data and the business logic.
- In the business tier, Enterprise JavaBeans (EJB) are defined. These beans provide the logic of the application and offer scalability by means of resource pooling. Session beans, stateless or stateful, perform a task for the client. Message-driven beans enable Java EE applications to process messages asynchronously. Web services, calling external services, can also be used in this tier.
- The persistence tier contains entity beans. Each entity represents a table in a relational database. Defining entities can be done by using the Java Persistence API. These entities will be used to persist the data in the databases.

A. Implementation details

The 3-tier architecture is depicted in Figure 3. The persistence tier contains all the entities, representing the tables in the database. Entities are annotated, plain java classes (POJO), that can be related to each other using multiplicities, such as one-to-one and many-to-one. The entity manager is part of the business tier. This service can be used to address the persistence tier and communicate with the database through the entities. The entity manager will also communicate with the synchronization component to sync with other database and remaining information systems, such as the electronic health record (EHR) of the patients.

The business tier manages all the service components. A service consists of a session bean with logic and/or a web service. The mail component is based on the JavaMail API, which is also included in the Java EE framework. This component is able to send mails to the physicians desktop or PDA. When a service needs to be scheduled or timed, a timer service, based on the EJB 3.0 Timer Services is utilized.

The web tier groups all the JSP pages. This mainly means pages for the configuration of parameters, for example the mail addresses of the physicians and pages for representing decisions and providing guidance during the treatment of patients. The execution developer interacts with the client. An UML diagram can be selected to be loaded into the system. If necessary, some configuration can be done in this developer.

V. ANTIBIOTIC DOSE GUIDELINE

In the following sections, the need for and usage of the antibiotic dose guideline are discussed, next to other existing antibiotic systems.

A. Necessity

Half of intensive care patients are experiencing a bacteriological infection, and over 70% of all patients are receiving antibiotic therapy [16]. The prescription of antibiotics in the ICU is of great importance, as any delay in adequate therapy increases mortality [17]. Excessive or inappropriate antibiotic use contributes to the emergence of drug resistance [18] which further leads to a delay in the institution of effective therapy. Furthermore, adequate dosing is of utmost importance, as underdosage can lead to treatment failure and emergence of resistant strains, while overdosage is costly and can induce organ toxicity [19]. Therefore, giving the correct type and dosage of antibiotics together with a fast response to the patients changing parameters constitutes the key challenge in optimized anti-infectious therapy.

B. ABDose

Ghent University Hospital is a tertiary care facility in Belgium. The ICU has a total of 56 beds and is accommodated with an information system with computerized physician order entry (CPOE). This system, however, does not support all the guidelines currently used in the hospital. The ICU is divided into a burn, pediatric, cardiac surgery, medical and a surgical unit, and overall, 4000 patients are admitted annually. The existing Intensive Care Information System contains electronic ICU patient records, but lacks a decision framework to follow advanced antibiotic guidelines according to renal function.

The antibiotic dose guideline gives an advice when the dosage of the recorded medication should be adjusted. The generated dosage recommendations are based on the patients creatinine clearance level during an 24-hour period and a
This CDSS provides support for drug-allergies, dosage, drug when alerts should be given. Providers could then respond written to identify medications, routes of administration and rules to implement a decision support system. Rules were better results. Implementation of alerts to notify physicians. This may lead with the application. This decision support system lacks the have to decide whether they want to synchronize their PDA to receive new information. A limitation of this approach is the fact that physicians themselves have to decide whether they want to synchronize their PDA with the application. This decision support system lacks the implementation of alerts to notify physicians. This may lead to better results.

A study [21] carried out to reduce medication error, used rules to implement a decision support system. Rules were written to identify medications, routes of administration and abnormal laboratory thresholds. Next, they had to determine when alerts should be given. Providers could then respond to the alert and decide to keep, revise or delete the order. This CDSS provides support for drug-allergies, dosage, drug interactions and guidance for medication-related laboratory testing. The alerts are shown while prescribing medications. This system doesn’t send reminders when the physician ignored the alert. If physicians would be automatically reminded if they didn’t respond to the given suggestions, this could signify an improvement offered by the platform.

Sain et al. [22] propose a platform for the real-time monitoring of patients staying at home. These patients can transmit data, concerning their health, to the doctors. A middleware is built to handle data processing and information sharing. This layer will be placed between the data source and the graphical user interface (GUI). To develop this application, J2EE (the former name of Java EE), the model-view-controller (MVC) pattern and Weblogic Server are used to increase reusability. This choice was made, based on the system architectures of OpenEHR and HL7. This paper indicates that Java EE is a suitable architecture to implement a guideline management system.

VI. Evaluation

The implementation of ABDose was evaluated to get more insights in the performance of the application. Therefore, the average loading time of the application front end, as shown in Figure 4, was tested as a function of the number of patient information entries to be displayed. Normally, the creatinine clearance is calculated once a day for this antibiotic dosage guideline, thus testing with a maximum of 100 entries per patient will suffice.

This evaluation was executed on a MacBook Pro (2.4 GHz Intel Core 2 Duo, 2 GB RAM, Mac OS X version 10.6.3) in Netbeans IDE 6.8 with Glassfish v3 application server. The loading times of the application front end are visualized in Figure 5. The time to load the application for a patient will increase, based on the number of patient information entries to be displayed. A request for the front end application for a specific patient with 20 creatinine clearance values will only take 400 ms. The throughput ranges from 0.9 to 3.5 per second, based on the number of information entries per patient. Next, the times were calculated to submit a decision in the front end application. This decision is an update of the information record and indicates whether the physician

C. Existing systems

In [20], the effect is studied of computer-based decision support (CDSS) systems on antibiotic consumption in the ICU. Handheld devices, such as a personal digital assistants (PDA), are chosen to provide clinicians with information. In this way, the decision support is incorporated into the workflow of the doctors. This device can also be used to prescribe medicines, access patient information, lookup practice guidelines etc. In this study a CDSS was designed to provide information to prescribers, such as unit details, antibiotic guidelines, data on resistance and susceptibility profiles and current data of the patient. The content was developed as HTML pages or was translated into HTML pages. JavaScript was used for active pages. Synchronization is used to update the PDA. This architecture uses two middle systems and is implemented particularly for the prescription of antibiotics. Physicians have to synchronize their PDA to receive new information. A limitation of this approach is the fact that physicians themselves have to decide whether they want to synchronize their PDA with the application. This decision support system lacks the implementation of alerts to notify physicians. This may lead to better results.

A study [21] carried out to reduce medication error, used rules to implement a decision support system. Rules were written to identify medications, routes of administration and abnormal laboratory thresholds. Next, they had to determine when alerts should be given. Providers could then respond to the alert and decide to keep, revise or delete the order. This CDSS provides support for drug-allergies, dosage, drug
TABLE I
FRONT END UPDATE TIMES WHEN A DECISION IS SUBMITTED.

<table>
<thead>
<tr>
<th>Number of entries per patient</th>
<th>5</th>
<th>25</th>
<th>55</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average loading time in ms</td>
<td>66</td>
<td>179</td>
<td>419</td>
<td>529</td>
</tr>
<tr>
<td>Standard deviation in ms</td>
<td>8</td>
<td>32</td>
<td>51</td>
<td>60</td>
</tr>
</tbody>
</table>

agreed or disagreed with the clinical decision support system. These results are shown in Table I. These results indicate that the clinical decision support system can be integrated into the physicians’ workflow.

VII. CONCLUSION AND FUTURE WORK

This paper describes a platform for automatically translating and executing guidelines. Java EE and a service-oriented approach were chosen to implement this platform. This offers a modular, distributed, multi-tier framework with separation of business logic and integration with other systems. Using UML as a well-known format for representing the guideline will close the communication gap between software developers and the medical domain and make it possible to translate guidelines in a working application. The antibiotic dosage guideline was used as a practical use case and an initial evaluation was performed.

Future research will focus on the statistical analysis to capture new knowledge about the impact on patient’s stay and physicians’ uptake. By analyzing the feedback information of the antibiotic dosage guideline, the guideline itself can be further optimized or reconfigured to new clinical practice. This can be done by extending the reporting component of the system with data mining algorithms. For example, certain patterns could be observed in the prescription of antibiotics, extraction of rules concerning dosage or creatinine clearance and information gathering concerning resistance. After using the guideline management system over a long period and applying the system for different advanced ICU guidelines in clinical studies, an analysis can be made as to what extent a decision support system will improve quality of care.

REFERENCES