AUTOMATED HEALTH BUSINESS PROCESS MODELLING AND ANALYSIS FOR E-HEALTH SYSTEM REQUIREMENTS ELICITATION

Ahmed H Alahmadi, Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia, a.alahmadi@latrobe.edu.au

Ben Soh, Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia, b.soh@latrobe.edu.au

Azmat Ullah, Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia, a.ullah@latrobe.edu.au

Abstract

e-Health is a rising research area in the intersection of public health and business sectors which refers to health and information services that are delivered or improved by using the Internet and other related information and communication technologies. Various aspects of e-Health are discussed in the literature; however, there is a research gap in the areas of system requirements engineering and health business process. It is widely accepted that health business process can play a pivotal role in capturing the details of e-Health system requirements. Therefore, understanding health business process is important to obtain suitable e-Health system requirements. A previous work proposed a methodology that manually models and analyses health business process for the aim of eliciting valid e-Health system requirements. However, the manual modeling and analysis technique is time consuming and may not be effective in the rapidly changing e-Health environment. This paper aims to overcome this limitation. The paper presents the automated approach of health business process modeling and analysis for e-Health system requirements elicitation. A case study on routine patient consultation visits in a healthcare organization has been used to validate the approach.

Keywords: Automated, Process Modelling, e-Health, Requirements elicitation.
1 INTRODUCTION

Expanding expectations from emerging e-Health systems means that extensive research is needed to address various issues in requirements engineering, specifically regarding health business processes, goals and objectives (Bliemel and Hassanein 2004, López, Massacci et al. 2009). Requirements engineering is one of the most effective ways of achieving successful development of an e-Health system that meets organisational expectations and helps to effectively reach business goals and objectives. Health business processes can play a pivotal role in capturing the details of e-Health system requirements, which is important to obtain suitable e-Health systems requirements.

The rapid development of the e-Health environment brings a serious challenge for e-Health system developers in modelling and analysing health business processes and goals to obtain system requirements that meet modern e-Health expectations against backdrop of continuous change. The manual modelling and analysis technique is time consuming and may not be effective in the rapidly changing e-Health environment. Generally, requirements engineering tasks in an environment like that of e-Health need to be automated to manage the rapid change of business processes, goals and objectives.

The previous paper (Alahmadi, Soh et al. 2014), proposed a methodology that manually models and analyses health business processes and goals for e-Health system requirements. It aimed to help e-Health system developers to better understand the e-Health environment and drive system requirements from the goals and activities associated with health business processes. However, one of the main limitations of that approach is that manual modelling and analysis do not support the dynamic characteristics of an e-Health environment. This paper aims to overcome this limitation.

Modelling health business processes automatically means using information systems to manage, tie together, coordinate, clarify and monitor the flow of the e-Health business sub-goals that the organisation wants to achieve. The e-Health systems support the task of automatic modelling, offering companies a solid platform for automation, referred to as the Business Process Management Suites (BPMS). This paper presents the automatic modelling and analysis of the health business process that was presented in a previous work. Two main elements were involved: automatically created UML sequence diagrams to resolve errors or conflicts among proposed health business processes and automatically generated UML state diagrams that represent a true picture of the health business process and its relationship to e-Health business goals, explaining the system’s aim and how it should be achieved.

2 AUTOMATED APPROACH OF DERIVING E-HEALTH SYSTEM REQUIREMENTS FROM HEALTH BUSINESS PROCESSES

In e-Health system development projects, as a complex information system, a great number of unstructured tasks and processes become available and must be analysed to transform them into structured requirements (Hong and Lingzi 2000). Requirements elicitation process is known to be time consuming and error prone when performed manually by a requirements engineer. This paper proposes an automatic methodology to elicit the e-Health system requirements from the health business process. The proposed methodology aims to model and analyse health business process automatically, to provide assistance to e-Health system developers to develop proposed systems according to current requirements driven by healthcare organization expectations. This will efficiently improve the optimisation level of extracted requirements by being process oriented and meeting modern e-Health business objectives.

The proposed methodology is divided into two tiers, as shown in Figure 1. The first tier includes healthcare business strategy specification and healthcare business infrastructure in a healthcare organisation. The concept of this tier was defined in depth in the previous work (Alahmadi, Soh et al. 2014). The second tier is the focus of this paper; it describes the automatic method of modelling and analysing health business processes to extract e-Health system requirements. Modelling and analysing health business processes to obtain system requirements is done in four phases. In the first phase, a
health business process is translated into a UML sequence diagram. This is the IT analyst’s task and needs to be done manually. The UML sequence diagram allows analysts to explore detailed information on the chosen business process. The second phase illustrates the sequence diagram data into a loosely coupled format called Extensible Mark-up Language (XML) metadata interchange (XMI) so that the information can be mapped; that is, so that similar elements can be grouped. XMI is an object management group approach for exchanging metadata through XML. The third phase involves analysing the business process to clarify and resolve conflicts between process-related activities (UML state chart in XMI format). Finally, in the fourth phase, the UML state chart is converted from XMI format into a graph to clarify the health business process in terms of e-Health system requirements for system developers.

![Diagram](image)

**Figure 1. Approach to modelling and analysing e-Health business processes automatically**

As system development must not be time consuming, the proposed methodology aims to provide e-Health system developers with an automated way of developing a suitable e-Health system that meets proposed goals and objectives. Automating the processes of the requirements-elicitation phase will clearly enhance the evolution of the e-Health arena because the criticality and complexity of this phase mean that processes are time consuming. This provides the capability to manage the rapid change that characterises e-Health environment in terms of processes and goals. Moreover, as healthcare business goals vary from one scenario to another and from one region to another, the automatic technique will allow system analysts to model and analyse health business process from different environments to extract the exact requirements and ensure their validation in different e-Health scenarios.

### 2.1 Tier 1: E-Health environment

The e-Health environment is highly changeable because of the variety of stakeholders, scenarios and ICTs involved. This variation of human and technical components creates a challenge in managing the goals and objectives behind the healthcare business. This characteristic means that goals associated with an e-Health environment are changeable and difficult to achieve. However, understanding the e-Health environment in terms of goals, sub-goals and activities that relate to different healthcare scenarios might be one of the best methods to manage health business process and goals to extract e-Health system requirements. Without a clear understanding of the e-Health environment and its goals, the function of eliciting e-Health system requirements is challenging.

Requirements elicitation corresponds naturally to the business process of the intended system. In recent years, business processes were introduced to software engineering and, in particular for early requirements and non-functional requirements modelling, business processes are now considered...
The importance of requirements elicitation as the first phase of the software lifecycle is becoming increasingly prominent. However, software analysts are specifically concerned with how fast and accurately system requirements can be derived/elicited and automating the process of requirements elicitation to improve time efficiency has long been their main goal. The focus of this has been on improving and partially automating the function of analysing and modelling business processes from different scenarios and environments, which must be done by a requirements engineer, and to generate mostly autonomous algorithms that extract requirements from a business process, as well as requirements compilation and patterns. In an e-Health system, where business processes and goals rapidly change, this aim has become critical.

The second tier of the approach contributes to this aim, as it proposes an automated e-Health requirements-elicitation approach based on a health business process. This tier involves a number of processing units (organised in a pipelined architecture) to analyse a complex health business process. The automated e-Health system requirements–elicitation tier separates the health business process into four main phases: translation, conversion, analysis and derivation. This produces the e-Health system requirements in the form of a UML state chart. The following sub-section discusses each one of these phases.

### 2.2.1 Translation of a health business process

To model and analyse a health business process for system requirements elicitation, the case study outlined in (Alahmadi, Soh et al. 2014) serves as the process in this paper. The case study was taken from the clinical scenarios available for healthcare professionals in the Australian e-Health learning centre (Australian Government 2012). Four steps are required to translate the chosen health business process into a UML sequence diagram, according to the proposed methodology:

- Identification of the business process tasks and sub-tasks.
- Identification of the participating objects.
- Identification of the message sender and receiver.
- Identification of alternative ways of sending and receiving messages.

In the first step, the chosen health business process is translated into a UML sequence diagram for the purpose of analysing and obtaining valid e-Health system requirements. This will allow the system developer to gain a true picture of the business process involved in the e-Health system. The second step aims to determine which stakeholders are involved in the business process, and the third step ensures accuracy between the developments of each component of the system by using the sequence diagram to introduce classes that work as parent methods, where messages travel between them. These classes may also be called the message sender, where the message is initiated, and receiver, where the message ends. In the fourth step, a unique identification number (ID) is assigned when a desired business process is to be converted into a message of the sequence diagram. Consequently, condition expression can be used to represent alternative ways of ending the message with the help of the message’s ID. The sequence diagram in Figure 2 shows how the process of a routine patient consultation visit to a healthcare centre was translated into a UML sequence diagram.
2.2.2 Conversion of a health business process from a UML sequence diagram into XMI format

To analyse the health business process that has been presented as a UML sequence diagram, it is critical to convert the process into a format that can be easily managed and compiled automatically. This step makes it easier to apply different conditions and distinguish between similar sub-tasks of the business process. After translating the health business process into a valid UML sequence diagram, this diagram must be converted into XMI format (Byrd and Turner 2000). This will simplify the analysis phase because, in XMI file format, each class or interaction message of the sequence diagram has a unique ID. In this study, the ArgoUML tool has been used to draw and convert the sequence diagram into XMI format. ArgoUML is an open-source application used to model UML diagrams and is implemented in Java and distributed under the Berkeley Source Distribution license. One of the main features of ArgoUML is that it is available on any platform supported by Java.

![Sequence Diagram]

Figure 2. The process of a routine patient consultation visit to a healthcare centre in a sequence diagram

2.2.3 Health business process analysis using XMI format

Business process analysis is used to obtain a systematic and detailed understanding of the system requirements, as defined in system project initiation and confined in the business case, and to reduce it into separate system requirements, which are then categorised, re-examined and decided on. The key goal of business process analysis is to generate a thorough functional specification of a system, defining the complete set of system abilities to be developed, along with supplementary data and procedure models that demonstrate the information to be supervised and the activities to be maintained by the new information system. The functional specification continues throughout the system development lifecycle and during the phase of system requirements identification. In the case presented here, for an analysis of the health business process, the methodology takes the health business process in the form of sequence diagrams, which have been generated as or converted into XMI. The methodology uses the Java Document Object Model (JDOM) to read the input XMI, line by line. JDOM uses XML parsers to construct documents. Simply, a Java demonstration and producer
of an XMI or XML document demonstrates a means to represent that document for ease of reading, treatment and writing, as required in the methodology. JDOM, developed by Jason Hunter and Brett McLaughlin, is common among Java developers’ documents. It comprises several methods, where each method carries a different role. However, here, only four JDOM methods are used to read the XMI file (see Table 1).

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SAXBuilder = new SAXBuilder ();</td>
</tr>
<tr>
<td></td>
<td>To build a JDOM document from the input file</td>
</tr>
<tr>
<td>2</td>
<td>Elementroot = docmnet.getRootElement ();</td>
</tr>
<tr>
<td></td>
<td>To get direct access to the DOM document root</td>
</tr>
<tr>
<td>3</td>
<td>Listrow = root.getChildren (“string”);</td>
</tr>
<tr>
<td></td>
<td>To give precise control over each related element in the business process</td>
</tr>
<tr>
<td>4</td>
<td>Stringname = attribute.getAttribute (“name”).getValue ();</td>
</tr>
<tr>
<td></td>
<td>To return an attribute value by name</td>
</tr>
</tbody>
</table>

**Table 1: JDOM methods and their function**

Once the e-Health business process has been converted into the sequence diagram, the health business process is analysed using Algorithm 1 in Table 2, which describes how the health business process is analysed by taking a valid XMI file as an input and using a JDOM parser to extract the necessary information. Based on the current value of JDOM, Algorithm 1 decides which information is extracted at the time. Since the XMI file represents the sequence diagram of the health business process, three main JDOM objects are expected: model, classifier role and interaction message. Each of these objects represents different information. For example, if the JDOM becomes equal to the model object, then the extracted data are the model name and ID, where the model represents the header of the XMI, which includes the XMI version. In the header, if JDOM is equal to the classifier role object, it extracts the classifier name and ID, where the classifier is the content of the XMI, which is used to represent the class name of the UML sequence diagram. If JDOM is equal to the interaction message object, it extracts the child elements of the classes, where interaction.message is the message between the two classes. However, to analyse the sequence diagram–related activities, identifying conflicts or errors between the classes and interactions of the sequence diagrams is essential. If any duplication is found among classes or interactions, the program resolves it automatically. In another words, let sequence diagram, which represents the business process, be represented as follows:

\[
D_0 \rightarrow D'_0, D_1 \rightarrow D'_1, D_2 \rightarrow \cdots \cdots \rightarrow D'_{r-1}, D_r \rightarrow D'_{r} \tag{1}
\]

In equation 1 the IM\(_i\) are interaction messages between sequence diagram classes and \(D_i, D'_i\) are the classes of the sequence diagram immediately before and after the interaction message IM\(_i\) is executed. The sender and receiver are denoted by \(M^\text{Sender of class}_i\) and \(M^\text{Receiver of class}_i\) respectively. \(D_i\) will be used as a notational expedient to indicate either \(D_i\) or \(D'_i\), \(D_i[K]\) is the \(K\)th element of the class \(D_i\). \(V_K\) will indicate the name of the message connected with position \(K\) in the interaction message. The initial sequence diagram classes are attained directly from the message specification. If \(M_i\) has pre-condition \(V_K = y\), then suppose \(D_i[K] = y\), and if \(M_i\) has pre-condition \(V_K = y\), suppose \(D'_i[K] = y\). Otherwise \(D_i[K] = D'_i[K] = \_\). Since each message among the sequence diagram classes is precise independently, the initial sequence diagram classes and its messages will contain a lot of unknown values. This can be done in two ways: First, amalgamation, two sequence diagram classes or interaction messages \(D_i\) and \(D_j\) (\(\neq j\)), are supposed to be the same if they are amalgamation, for instance there exists a elements assignment \(\emptyset\) such as \(\emptyset(D_i) = \emptyset(D_j)\). Second, the frame axiom, for each \(j\), if \(D_i[j] = \_\), then suppose \(D_i[j] := D_i[j]\) and if \(D_i[j] = \_\), then suppose \(D'_i[j] := D_i[j]\).
1. Begin
2. Variables:
   Read "Health business process": JADOM;
   Identify the convert version of XMI: XMI-Model;
   Identify classes in "Health business process": interactionRole;
   Identify messages among the process **: interaction_messages;
   Vs : valide Health business process in the form of sequence diagram;
3. Body
   Let Vs : \{Vs1; Vs2, Vs3, ..., VsN\} is a valid sequence diagram in the form of XMI;
   For each row/line Ri of the sequence diagram Vs Ri ∈ Vs do
      If (JADOM == classifierRole) then
         obtain the classifier_Name, classifier_ID;
      else if (JADOM == Model) then
         obtain model_Name, model_ID;
      else if (JADOM == interaction_message) then
         obtain interaction_Name, interaction_ID;
   end If;
   end for;
   // Resolve Error and Conflicts
   For each classifier of sequence diagram Vs and interaction_message of Vs, do
      If (New classifier_Name == Previous classifier_Name) then
         state error "duplicate class in the sequence diagram";
      else if (New classifier_Name == Null) then
         state error "there is zero class Name";
   end If;
   // similarly check interaction_message
   end for;
   For each sequence diagram of Vs, I ∈ Vs do
      temporary.ID <= I.ID;
      if (I = = classifier.ID) then
         obtain message sender and receiver;
      end if;
   end for;

Table 2: Automated goals analysis

This assumes, of course, there is no unknown side effects amongst the messages of the sequence diagram classes. These two approaches allow the business and technology analysts to detect conflict between the sequence diagram classes and its interaction messages. A conflict is detected if the sequence diagram contains duplication classes or interaction messages. Any accounted conflicts must be resolved by the technology analysts and the algorithm is started again. Once the business process has been finalized, it is then translated into the UML sequence diagram for simplification, as shown in Figure 3. This diagram depicts one business process in three different scenarios, each scenario representing one sub-goal of the process and where each class or message of the sequence diagram has been signed a unique XMI ID. At this stage, the business process is clear and is ready to translate into UML state chart.
2.2.4 Derivation of e-Health system requirements (elicitation)

Modelling and analyzing health business process manually is time consuming task which alternately negatively influence the development of accurate e-Health system that fulfill the expectations of all stakeholders on time and completely. After the health business process has been finalised and analyzed using UML sequence diagram, the process then converted into the UML state charts. UML state chart describe what need to be included in the health business process in the viewpoint of e-Health system developers. Algorithm 2 of Table 3 describes how the methodology converts sequence diagram into state charts. The state charts at figure 4 gives a true picture of the proposed health business process and e-Health system requirements elicitation. Figure 4 shows how the proposed methodology converts the process of a routine consultation visit to healthcare center from UML sequence diagram to UML state charts. Figure 4 at this stage presents three different types of state charts where each state chart represents different scenario of the proposed health business process.

Figure 3. XMI of the health business process in sequence diagram
1. Begin
2. Variables:
   3. Read "Health business process": JADOM;
   4. Identify interaction in the sequence diagram: interaction_message;
5. Body
6. Let File_Out be the valid sequence diagram;
7. If File_Out is not exist then
   8. Creat a new file;
9. end if;
10. Let Vs : {Vs1; Vs2, Vs3, ..., VsN} is the valid sequence diagram;
11. For each row/line Ri of the file Vs Ri ∈ ViN do
12.   If (JADOM == interaction_message, i) then
13.      temporary.value <= i.value;
14.      temporary.name <= i.name;
15.      obtain sender value (SE) and receiver value (RE);
16.   end if;
17.   if (i.value = = SE) then
18.      write as state of the state chart in File_out;
19.   else if (i.value = = RE) then
20.      write as label of the state chart in File_out;
21.   end if;
22. end for;

Table 3: Automatic generation of UML state chart

First scenario, starts once the healthcare center system registers the patient, the system then check the patient details for example health insurance details and appointments. Also the healthcare system checks the availability of medical staff based on the condition of the patient. In the case of unavailability of medical staff, the system then automatically advice patient with the available next appointment for the suitable doctor through the administration actor, but if the doctor is available the consultation then go further for completion.

Second scenario, starts once the system confirm the availability of medical staff to assist patient, the system then forwards patient information to the doctor for the clinical examination. During the consultation time doctor has the capability to access patient medical record stored in the centralized medical record database. This will enhance the ability of the doctor to understand the exact condition of the patient through the historical information stored in the medical record database. After this stage, doctor can assist patient with the suitable medication according to his/her condition. Prescription will be available then in the medication department, where pharmacist can also access the medical record database to check the medication history of the patient.

Third scenario, starts once the doctor decides that patient needs to undertake some further examination. In some health condition doctor advice patient to see other health professional or do some examination such as blood test or x-ray. In this case, doctor refers the patient through the system to other departments by uploading the required examination or notes. The health professional or medical staff in the examination department also can access patient health record to view his/her medical recorded on the shared health summary uploaded by one of his/her previous regular doctors. The final stage of all scenarios is the administration state, where patient is discharged and follow up visit is organized.

The state chart diagram at this stage clearly answers the following questions: how to define the health business process and its belonging business goals priory to derive the e-Health system requirements from them? How to develop a better understanding of how health business process is functioning? Does managing continuously improvement in health business process is possible? And finally how to get new health business process activities defined and into operation quickly and easily? The answer
of all these questions positively influences the method of deriving e-Health system requirement from the e-Health environment particularly business process which alternately positively influences the development of e-Health system according to the healthcare business expectation and influences the alignment process between business and IT.

**First Scenario**

**Second Scenario**

**Third Scenario**

**Legend**: ○ Initial State  ● End State  □ State

*Figure 4: XMI of e-Health system requirements in the form of state chart*
To summaries and combine the four steps involved in the proposed methodology, Figure 5 present a simplified case of Remote Patient Monitoring Process (RPMP). Figure 5.a represents a simple sequence diagram for typical Remote Patient Monitoring Process (RPMP). The sequence diagram represents three primary actors: the patient, the Remote Monitoring Management System (RMMS) that operates the RPM solution and an external HCP system. Typically, the RMMS checks patient’s vital signs; for example, blood pressure or glucose, several times per day. Once the RMMS aggregates the vital sign assessment data, it compares the values against a pre-established reference range or threshold to identify those readings that are out of normal range. If necessary, a notification is generated to the HCP system, notifying medical staff of the abnormal readings. Both normal and abnormal information is transmitted to the HCP system. The medical staff review the information, make clinical notes, generate referrals and modify treatment plans. The updated treatment plan is transmitted to the RMMS and subsequently communicated to the patient and patient local devices.

The ArgoUML tool allows the user to export the XMI of the desired sequence diagram with clear identification of the sequence diagram classes and interaction messages. Figure 5.b, for example, shows the XMI of the RPM sequence diagram presented in Figure 5.a. As shown, the program separates the classes, interaction messages and senders and receivers of the sequence diagram and assigns them a unique ID. For instance, the RPM sequence diagram at Figure 5.b contains three classes: patient, RMMS and HCP system, with their unique IDs being0000000000000CDF, 0000000000000CE2 and 0000000000000CEB. Similarly, Figure 5.c shows the sequence diagram classes related to messages and their unique IDs. Figure 5.d shows how the RPM sequence diagram is converted from XMI format into a UML state chart graph.

Figure 5. The proposed approach steps for Remote Patient Monitoring Process (RPMP)
3 CONCLUSION

In relation to the requirements engineering concept in the e-health domain, it is widely accepted that health business processes can play a pivotal role in capturing the details of e-Health system requirements. This paper has presented a technique to model and analyze health business process and associated business goals automatically, enabling system developers to developed suitable e-Health system. The technique is categorized into four steps. Step 1 translates health business process into a UML sequence diagram which identifies the healthcare business goals. Step 2 shows the conversion of health business process from a UML sequence diagram to XMI set-up so that it can be analyzed. Step 3 presents the conversion of the health business process from a UML sequence diagram into a UML state chart in XMI set-up. Finally, step 4 presents the conversion of the state chart from XMI set-up to graph set-up.

The proposed methodology provided e-Health system developers with an automated way of developing a suitable e-Health system that meets proposed goals and objectives. It enhanced the capability to manage the rapid change that characterises e-Health environment in terms of processes and goals. Moreover, as healthcare business goals vary from one scenario to another and from one region to another, the automatic technique will allow system analysts to model and analyse health business process from different environments to extract the exact requirements and ensure their validation in different e-Health scenarios. The proposed methodology also provide a method of resolving errors or conflicts among health business processes and automatically generated UML state diagrams that represent a true picture of the health business process and its relationship to e-Health business goals, explaining the system’s aim and how it should be achieved.

References


