Goal-Driven Development of a Patient Surveillance Application for Improving Patient Safety

Saeed Ahmadi Behnam[1], Daniel Amyot[1], Alan J. Forster[2], Liam Peyton[1], and Azalia Shamsaei[1]

[1] SITE, University of Ottawa, 800 King Edward Ave
[2] Department of Medicine, University of Ottawa and OHRI
Ottawa ON, K1N 6N5 Canada
sahma088@uottawa.ca, damyot@site.uottawa.ca, aforster@ohri.ca, lpeyton@site.uottawa.ca, asham092@uottawa.ca

Abstract. Hospitals strive to improve the safety of their patients. Yet, every year, thousands of patients suffer from adverse events, which are defined as undesirable outcomes caused by health care business processes. There are few tools supporting adverse event detection and these are ineffective. There is hence some urgency in developing such a tool in a way that complies with the organizations goals and privacy legislation. In addition, governments will soon require hospitals to report on adverse events. In this paper, we will show how a pilot application we developed contributes to the patient safety goals of a major teaching hospital and how our goal-driven approach supported the collaboration between the university researchers and hospital decision makers involved. Benefits and challenges related to the modeling of requirements, goals, and processes, and to the development of the application itself, are also discussed.

Keywords: Adverse Events, Business Process Modeling, Goal Modeling, Health Care, Patient Safety, User Requirements Notation.

1 Introduction

Modern health care is a data- and knowledge-intensive enterprise. Information technology (IT) systems are increasingly used in health care organizations to collect, analyze, manage, and share information and knowledge. Although one of the main goals in this industry is to improve quality of care, IT systems are often not aligned with this primary goal. According to a recent report from the US National Research Council [11], in which the authors studied eight medical centers acknowledged as leaders in their usage of IT, such systems in health care are used in practice more for regulatory compliance and lawsuits protection than to improve clinical care.

Patient safety is one important sub-goal of health care quality, and minimizing the number and severity of *adverse events*, which are undesirable patient outcomes caused by medical care, contributes greatly to patient safety. It is not only important for hospitals and other health care organizations to define and support processes for detecting, assessing, and reporting on adverse events, but, in fact, this is being turned into a legal obligation in many provinces and states.

G. Babin, P. Kropf, and M. Weiss (Eds.): MCETECH 2009, LNBIP 26, pp. 65–76, 2009.

Often, paper-based approaches are used to support such processes, and they may vary from department to department. In this context, there is both a need and an opportunity to take advantage of e-technologies to improve the efficiency and effectiveness of existing health care processes. However, it has been observed that current IT applications in this area tend to "simply mimic existing paper-based forms and provide little support for the cognitive tasks of clinicians or the workflow of the people who must actually use the system" [11].

This paper reports on our experience and lessons learned during the development of a *Patient Surveillance* application targeting the detection of adverse events. This project is a joint venture between health care professionals from The Ottawa Hospital (TOH) and researchers from the Ottawa Hospital Research Institute (OHRI) and the University of Ottawa. This tool supports a prospective surveillance process in order to improve the accuracy of adverse event detection (and hence improve patient safety) while minimizing its associated costs.

The development approach taken is driven by the goals of the organization and other stakeholders, in order to avoid repeating the same mistakes identified for existing IT systems as discussed above. It combines state-of-the-art requirements engineering techniques and e-technologies. Requirements (e.g., goals, processes and database schemas) are elicited using a combination of models in the User Requirements Notation (URN) and UML. The main project objective, improving patient safety, was decomposed into four sub-goals: data collection, information generation, knowledge creation and knowledge application. Goal and process models were created for all of them, but the scope of the first phase of this project was limited to the first two goals.

A Web-based application was created and then used by a nurse to monitor patients using a mobile tablet PC for a one-month period (so far), and by physicians to assess whether the observations were indeed adverse events, with probable causes.

The rest of the paper is as follows. Section 2 provides background information on adverse events and on the notation used to model the target business process and its goals. In section 3, we describe our development approach, which is then detailed with the business process and the implementation of the surveillance application itself in section 4. Observations and lessons learned are discussed along the way. Finally, the last section provides conclusions and opportunities for future work.

2 Background

2.1 Adverse Events in Health Care

Adverse events are undesirable patient outcomes caused by medical care rather than the underlying disease process [12]. An example of an adverse event is an allergic reaction caused by a medication. The reaction would not have occurred if the patient had not been exposed to the medication. In most instances such events are not avoidable. However, in a substantial proportion, they are preventable as they are due to an error. For example, if the prescribing physician neglected to enquire about prior allergic reactions to medications when she prescribed the medication, then the patient may be exposed to a harmful medication unnecessarily.

Unfortunately, there are large numbers of adverse events in the health care system. Focusing specifically on hospital patients, Canadian studies estimate that one in twelve hospital patients experience an adverse event [2]. A third of these adverse events are preventable. More importantly, one in six patients dies as a result of the adverse events. Extrapolating this risk to the Canadian population of hospital patients, there are 28,000 deaths annually due to medical errors. While this statistic is alarming, the risks are probably greater across the entire health system, which includes institutional care and ambulatory care. Both of these settings are also associated with an important risk of preventable injury [5, 6].

The current health care industry has immature systems to detect and monitor adverse event occurrence. The accepted method of adverse event detection is voluntary incident reporting. The method does not identify over 90% of adverse events [6]. Despite this fact, it is mandated by most accrediting bodies. More sophisticated methods of adverse event detection have been tested and are in development, including two-stage chart review, administrative data surveillance, electronic health record surveillance, and clinical surveillance [9].

Prospective adverse event surveillance holds promise as method [4, 9]. In this approach, a nurse monitors patient care for pre-specified triggers. When they occur, specific information is recorded and a case summary is generated. Case summaries are reviewed by physicians to determine their importance. This is a very cost-effective approach, even when considering the nurse's salary. The approach was developed based on prospective surveillance experience in 5 hospital units. These units included: a general medical unit, an intensive care unit, a cardiac surgical intensive care unit, an obstetrical unit, and an orthopedic surgery unit. The general approach for identifying adverse events worked effectively in all units despite there being very different patients, work processes, and adverse event types identified.

Although this method identifies more adverse events than other techniques, there is a need to develop IT infrastructure to support its activities. Because the general approach is modified slightly for each unit as different patient characteristics are measured and different adverse event triggers are monitored, supporting software must be built to accommodate customization.

2.2 Business Process Modeling with the User Requirement Notation

The *User Requirements Notation* (URN) is a graphical modeling language recently standardized by the International Telecommunication Union [7]. URN is intended for the elicitation, analysis, specification, and validation of requirements. URN allows software and requirements engineers to discover and specify requirements for a proposed system or an evolving system, and analyse such requirements for correctness and completeness.

The applicability of URN goes beyond requirements models; URN is also suitable for the modeling and analysis of business goals and processes [10, 14]. URN is composed of two sub-notations: the *Goal-Oriented Requirement Language* (GRL) for goal modeling and *Use Case Maps* (UCM) for scenario/process modeling.

GRL enables business analysts and IT architects to model strategic goals and concerns using various types of intentional elements and relationships, as well as their stakeholders called actors (⬭). Core intentional elements include goals (⬭) for

functional requirements, softgoals (⬭) for qualities and non-functional requirements, and tasks (⬭) for activities and alternative solutions. Intentional elements can also be linked by AND/OR decomposition and by contributions. Positive contribution levels may be sufficient (+), insufficient (+) or some positive (+). Similar levels exist for negative contributions (−). Quantitative contributions on a [-100, 100] scale may also be used. GRL *strategies* enable modelers to assign initial satisfaction values to some of the intentional elements (usually alternatives at the bottom of the graph) and propagate this information to the other elements through the decompositions and contribution links. This ultimately helps assess the impact of alternative solutions on high-level goals of the stakeholders involved. Such models are also useful for evaluating trade-offs, documenting rationales for design decisions, and modeling legal requirements.

Use Case Maps (UCM) are used to model scenarios and processes in the form of causal relationships linking responsibilities (×) which may be assigned to components (□). Responsibilities represent activities performed in a process whereas components represent actors, systems, and system parts. UCM support most of the concepts used in common workflow modeling notations including start points (●), end points (▌) as well as alternative and concurrent flows. Stubs (◇) are containers for sub-maps and can be used to organize a complex model in a hierarchical structure.

In our project, we used the jUCMNav open source software, an Eclipse plug-in used for creating, analyzing, and managing URN models [13]. This tool also supports extensions to URN for modeling key performance indicators (KPI) in the context of business process analysis and monitoring and performance management [10].

3 Highlights of the Development Approach

The approach selected is described below. The goal and scenario modeling part is inspired from the process proposed by Liu and Yu [8]. Several micro and macro-iterations were performed along the way.

1. **Stakeholder and goals:** Model, with GRL and jUCMNav, the stakeholders and their main high-level goal. Decompose the goals of the main stakeholder, namely The Ottawa Hospital (TOH) in our project.
2. **Alternatives and strategies:** Model the alternative surveillance methods as tasks and their contributions to TOH's goals. The comparison among these methods, enabled by computing the results of GRL strategies (automatically done with jUCMNav), was shown to the domain experts at TOH to ensure they complied with the result of their experiments with different methods [9].
3. **Processes:** Add UCM-based processes to the model, which realize the goals by satisfying the tasks mentioned in the goal view. New goals are often discovered along the way, so goals and processes can be aligned.
4. **Scope:** Evaluate risks and select a subset of goals and processes for the application. The scope was set to the prospective surveillance solution and was supported by goal and process models, clinical experiments, and constraints of our team (i.e., very busy physicians and part-time development by a graduate student with little experience in the selected technologies). Having GRL for modeling the goals and UCM for the processes in separate layers makes solutions independent from

deployment structures and early commitment to architectures. This also increases the reusability of the model in different environments (hospitals, departments) and increases its flexibility and maintainability when requirements change.

5. **Implementation:** Use conventional software engineering and (Web-based) development methods to implement the application. The latter, in our context, uses a 3-tier architecture to increase the maintainability and usability of software assets.
6. **Pilot validation:** Use the application in a pilot study. Ours was tested by an observer nurse at TOH's Clinical Teaching Unit (CTU) for a month to collect data and solve usability and deployment issues. At the time of writing this paper, reviewer physicians have just started using the application for reviewing the data collected and extracting adverse event information, which has value to decision makers. The nurse is currently involved in a second pilot, this time for 3 months.

4 Patient Surveillance Application

This section provides several details on the process steps from the previous section as well as several observations and lessons learned.

4.1 Defining the Goals and Evaluating Strategies

We first considered the main stakeholders of the patient surveillance application, including The Ottawa Hospital (TOH), physicians, nurses, decision makers, patients, health care government agencies, and university researchers. As their goals are directly or indirectly influenced (sometimes in conflicting ways) by the use of this application, GRL diagrams were created (step 1 in section 3). Being the primary care provider, TOH was the most influential stakeholder in the definition of the scope of the application. Fig. 1 provides a very partial view of our GRL model[1] and shows some of TOH's high-level goals and their relations to government goals.

Having such a goal model at this stage helped us understand the expectations of different stakeholders and how they interact. A few counter-intuitive relations were also observed, e.g., that improving patient safety had a positive contribution on cost reduction because of overall decreased lawsuits and patient care costs (as explained in the connected GRL belief (ellipse) in Fig. 1, which acts as a comment). The model was also useful to understand the scope of the project and its risks. Improving patient safety is the high-level goal of TOH that is targeted by the surveillance application.

In a prospective surveillance method, improvement of patient safety is done by collecting data about adverse events, having it analyzed by knowledgeable reviewers (physicians), and making decisions on how to improve patient safety by decreasing the possibility of adverse event occurrence (e.g., by improving an existing health care business process). However, there are different ways of addressing each of these steps, and each has positive and negative contributions on the stakeholder goals. These were also modeled in GRL (not shown here) and GRL strategies were defined.

[1] Our current (and evolving) model is comprised of 87 GRL intentional elements and 10 GRL actors part of 12 GRL diagrams, of 72 UCM responsibilities and 14 UCM components part of 21 UCM diagrams, and of hundreds of relationships between these model elements. From our experience, this is an average-sized URN model.

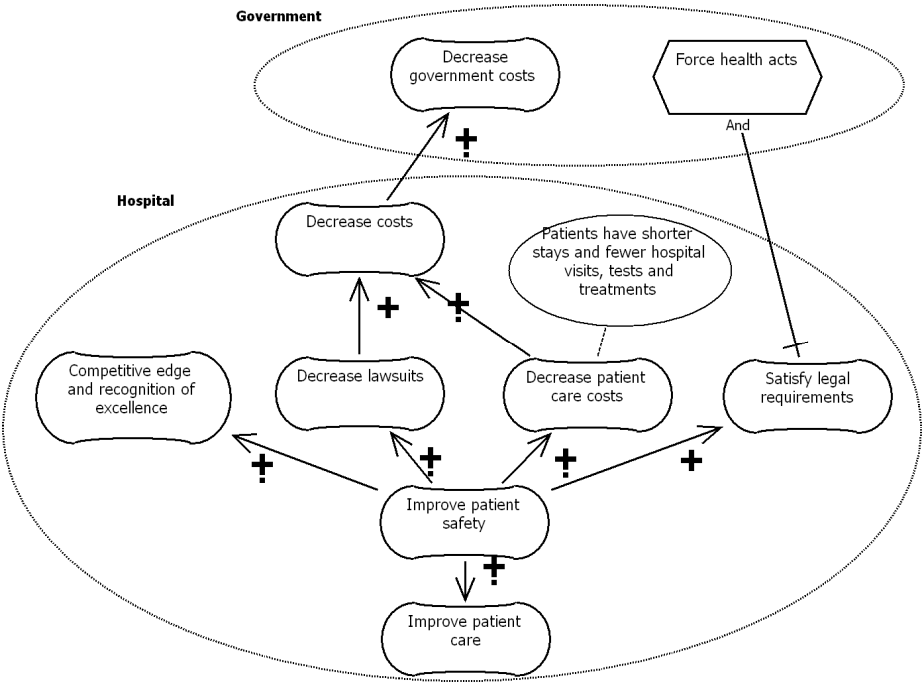

Fig. 1. Partial goal model from the TOH viewpoint

GRL models helped us reason about the requirements for patient safety. We have found that goal models are useful to communicate with stakeholders, especially domain experts, and discuss their requirements while conveying our own software engineering concerns. We used jUCMNav for comparing different alternatives by creating GRL strategies for each of them and then examining how they impact stakeholder objectives (step 2 in section 3). The visual evaluation feedback (GRL intentional elements become color-coded during an evaluation) helped stakeholders understand and assess such impact at a glance.

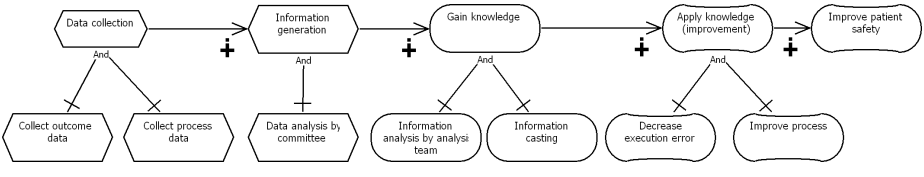

Fig. 2. Goals and high-level tasks of the prospective surveillance approach

Fig. 2 shows how the main goal of the TOH can be achieved through a set of softgoals, goals and tasks. This sequence starts with data collection and ends with applying the knowledge of how to decrease the adverse events. The whole sequence will

result in improving patient safety. These sub-tasks and sub-goals are then considered, refined, assessed, and realized in more detail in the next development stages.

Having the goal model at such an early stage provided an opportunity to understand similarities between adverse event detection methods. Although we first focus on proactive surveillance (as this is the most cost-effective method) our application can be made flexible enough to support other and complementary methods at very little cost. For example, we recently received a request to consider voluntary incident reporting (where a physician reports a potential incident instead of an observer nurse) as an addition. This only affects the Data collection task of Fig. 2, and modifications to the model and its implementation can then be localized to small parts only.

4.2 Modeling the Process Satisfying the Goals

With UCM, we then model a process that satisfies the combination of intentional elements selected from the goal model (step 3 in section 3). This UCM view refines the goal model by sequencing tasks and providing additional details in a workflow-like representation that would be otherwise cumbersome to capture. This also paves the way towards architectural descriptions and the support of specific use cases.

Fig. 3 gives a high-level view of the overall adverse event management process.

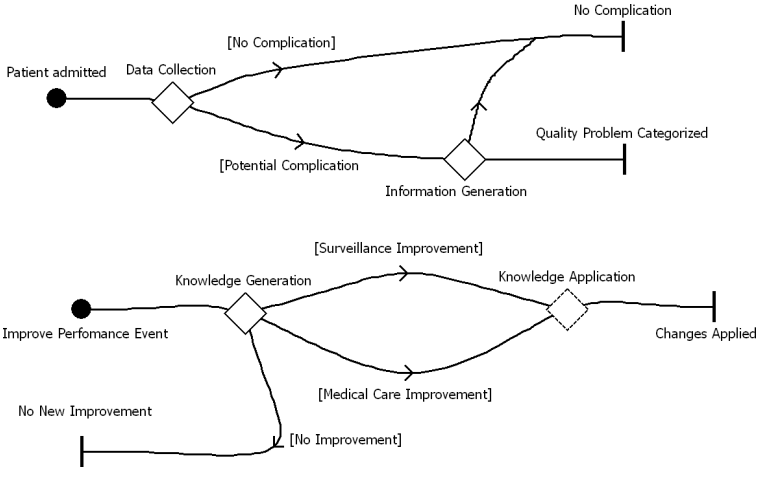

Fig. 3. High-level adverse event management process UCM

Stubs encapsulate the details of the sub-processes defining the four important intentional elements identified in Fig. 2, namely Data Collection, Information Generation, Knowledge Generation and Knowledge Application. This process view is independent of the underlying method of implementing each step. Also, UCM models offer the possibility to describe alternative process refinements with dynamic stubs (dashed diamond symbol). For instance, we have specified several possible ways of performing Knowledge Application, which are not shown here due to space constraints.

We have created UCM diagrams for all the stubs in Fig. 3. However, the implementation of the bottom half has been postponed to a second phase of the project

because of evolving requirements (we wanted to learn from the pilot study first) and the availability of development resources (step 4 in section 3).

As an example, Fig. 4 shows UCM diagrams detailing the Data Collection stub from Fig. 3, at three levels of abstraction. Part (a) is connected directly to the Data Collection stub and, given our focus on the prospective surveillance method, indicates that an observer nurse is involved. The Locate Health Care Quality Problem stub is refined in part (b), where the various responsibilities for observing processes and patient statuses are identified. As shown, many of them can be done in parallel or in any order. The three stubs in diagram (b) all contain the same diagram in part (c), which shows that sub-processes can be reused in many locations.

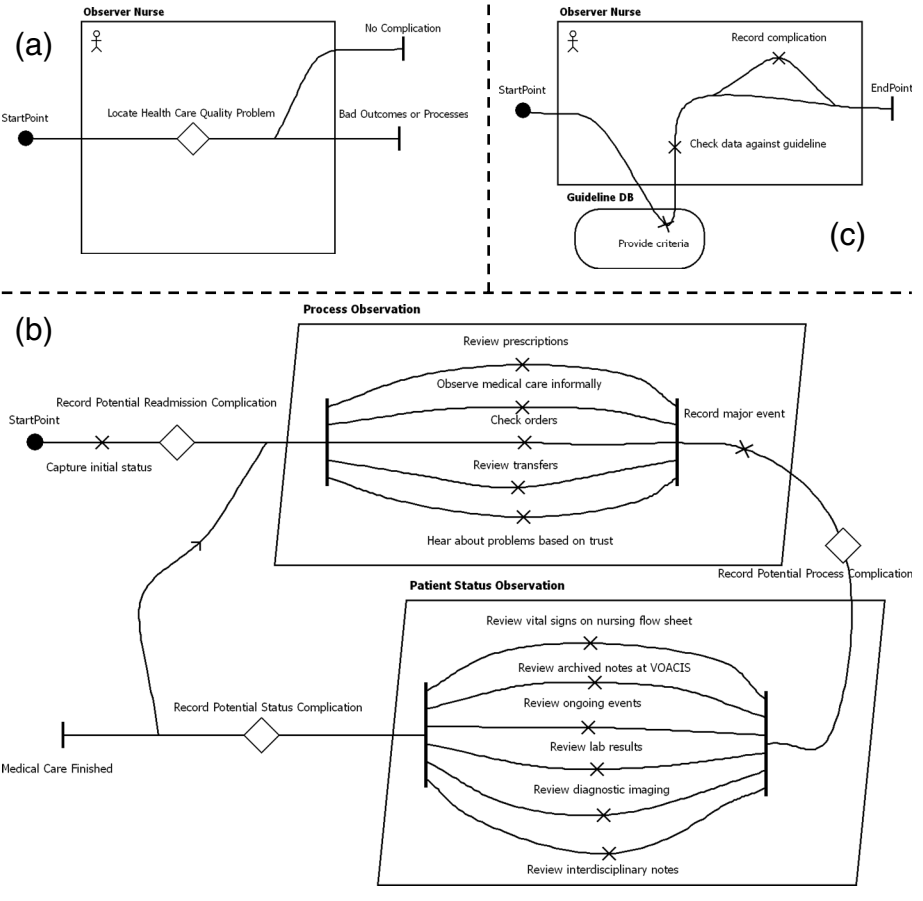

Fig. 4. UCM diagrams for prospective surveillance based data collection

Such models were useful when communicating with domain experts, but also with developers, which are more accustomed to use case models. Also, traceability from UCM elements to the GRL view helps them understand "why" the use cases are as

they are. UCM responsibilities can also be reassigned easily to other components along the way simply by dragging a responsibility and dropping it in the desired component box in a diagram, with jUCMNav. The cost of considering variations and of doing changes to the use case is then very low. Also, having different levels of details with sub-maps helps maintain the model when requirements change as it is possible to modify sub-models without breaking the general solution.

4.3 Software Architecture and Implementation

Considering the stakeholders' goals, requirements, and constraints, it was decided to use a Web application with a typical and loosely-coupled three-tier architecture to support the application (step 5 in section 3). This architecture is composed of a Web browser (on a tablet PC connected through a wireless network), a Web server containing a presentation layer (in ASPX) and a business logic layer, and a database server containing patient information and stored procedures. Different actors/roles (e.g., observer, reviewer, facilitator, and administrator) are given different tasks and access privileges. Constraints from TOH (who will eventually take over the maintenance of the application) included the use of the .NET framework and of MS SQL.

In our context, a Web application enabled the use of interfaces generally known to users, many of whom are very busy and require remote access, and eased application deployment. A central database enables the sharing of information across different users and across different steps of the business process.

We designed the database schema to support the goals and processes which were modeled in previous steps, also considering the types of requests users of future steps of our business process (the bottom part of Fig. 2) would likely perform. A UML class diagram was used to formalize the information about departments, patients, physicians, diseases and health problems, adverse events categories, observations, review decisions, etc. as well as many relationships such as who is in charge of a patient after admission.

To illustrate the interface, Fig. 5 shows the Web page that corresponds to the "Capture initial status" responsibility of the UCM in Fig. 4. An observer nurse uses this page to add patient data (some of which might eventually be obtained from existing operational system) and information about the care unit and physician to which the patient is assigned after admission. Many such pages were created for the various tasks and roles we identified.

4.4 Obstacles and Mitigations

Two major obstacles were encountered close to deployment time:

De-identification: A late requirement was added to satisfy health care privacy legislation and get permission to deploy the application for the pilot study. Identification information (e.g. name and patient identifier) needed to be stored behind the firewall of the hospital whereas the rest of the information needed to be on our database server in the research institute, behind a different firewall. The URN model was slightly evolved to reflect this requirement, and the code was changed to store the identifiers in a separate (XML) file instead of in the database. This file could then be located on

a different server, inside the hospital. Synthetic identifiers were generated and enabled the authorized users to access the information from both sources transparently and to present it in a combined way. However, this issue caused delays in the application deployment as well as stressful situations among stakeholders that could have been prevented by a more precise deployment plan.

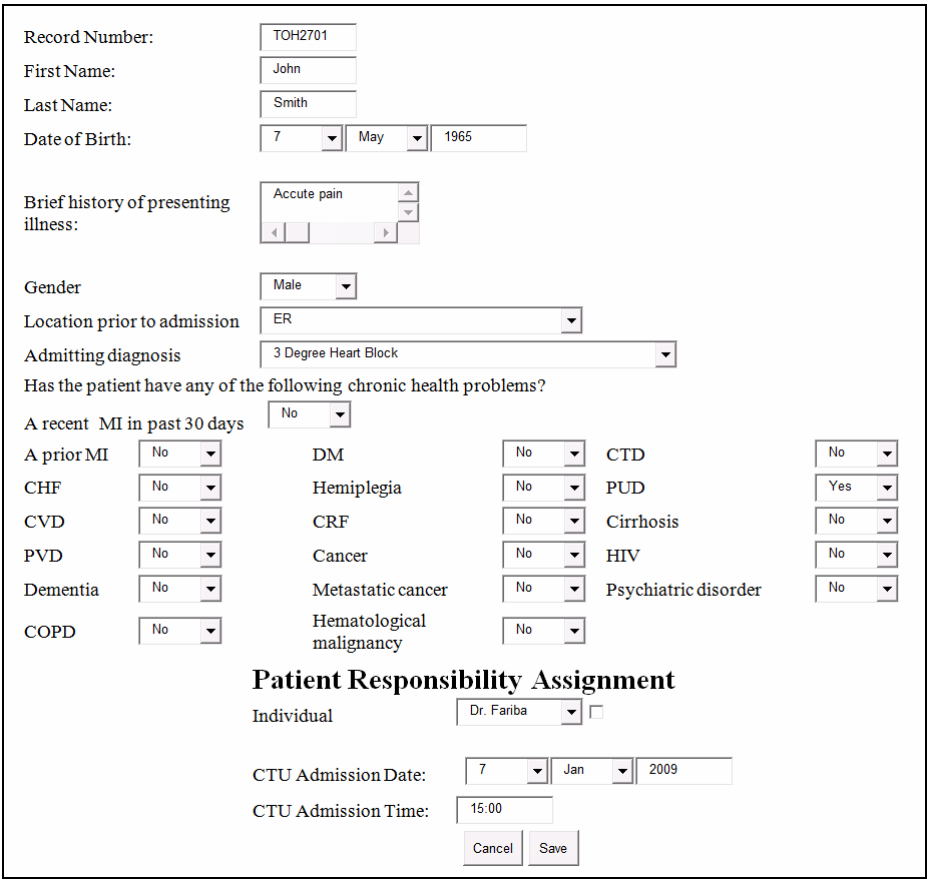

Fig. 5. Snapshot of a data collection Web page

Downgrading the data layer: Initial requirements targeted the MS SQL Server 2005 database management system for the application's data layer because the hospital planned to move from an earlier version (2000) to this one by deployment time. Our database was therefore created for version 2005, which is not backward compatible with version 2000. However, the upgrade to version 2005 was not available by deployment time. Our own server at the research center had version 2000, which was required by a companion Business Intelligence tool (Cognos 8). It was hence necessary to change some part of the data tier, and its separation from the business logic and presentation tier proved very useful at this stage as the remained untouched. Yet,

this introduced additional delays and stress as we spent time converting the data, tables and stored procedures to make them compatible with MS SQL 2000. This obstacle is an example of changes that can happen to all applications developed for industrial organizations by academic researchers. Continuous communication and visibility can decrease the risks related to unexpected changes to plans, and flexible architectures are essential to such collaborative projects.

5 Conclusion and Future Work

Joint projects and close collaboration between computer scientists and health care professionals are highly recommended by [11] in order to solve IT issues in this challenging area. This paper reported on an ongoing project targeting patient safety through the detection and analysis of adverse events, which can lead to the evolution of health care business processes. We have taken a goal-driven approach based on URN models that provided a suitable level of abstraction for productive university-industry collaboration in IT, where many stakeholders are busy and have very different background knowledge. Capturing and referring to "why" aspects helps to reduce the risks of misunderstandings, although this does not prevent all conventional obstacles (such as unexpected changes to deployment plans, see previous section) from happening. Such models are also resistant to change (because of their high level of abstraction) and flexible in case of changes (given their structure). They also provided design guidance for the later development steps, which led to a working application that takes advantage of e-technologies where paper-based approaches are often used. Other approaches based on goal models exist (such as van Lamsveerde's [15]) but they seldom combine goals with more detailed processes or use cases as well as what is possible with URN, which for now also contains the only standard goal notation.

The results of the pilot study (step 6 in section 3) are very encouraging so far and the experiment has been renewed for a 3-month period. Few adjustments had to be made to the prototype, yet we plan to improve it on various quality aspects such as usability, robustness, security and scalability. We also plan to deploy it in other hospital departments and even in a different hospital, where the culture, regulations and business processes are different. We expect the URN models to be quite reusable (given their generality) and the application itself should easily adaptable since we have made it customizable (in terms of departments, types of diseases, types of adverse events, etc.) from the beginning. We also plan to evolve the URN model to take advantage of KPI extensions proposed in [10] and use it for performance modeling.

Acknowledgments

This work was supported by a Collaborative Health Research Project grant from CIHR and NSERC (Canada) on *Performance Management at the Point of Care: Secure Data Delivery to Drive Clinical Decision Making Processes for Hospital Quality Control*. We are thankful to the OHRI personnel and to the following students for their help in developing and deploying the surveillance application: J. Blais, R. D'Angelo, M. Garzon, and R. Bougueng Tchemeube.

References

1. Amyot, D.: Introduction to the User Requirements Notation: Learning by Example. Computer Networks 42(3), 285–301 (2003)
2. Baker, G.R., Norton, P.G., Flintoft, V., Blais, R., Brown, A., Cox, J., et al.: The Canadian Adverse Events Study: the incidence of adverse events among hospital patients in Canada. CMAJ 170(11), 1678–1686 (2004)
3. Cullen, D.J., Bates, D.W., Small, S.D., Cooper, J.B., Nemeskal, A.R., Leape, L.L.: The incident reporting system does not detect adverse drug events: a problem for quality improvement. Joint Commission Journal on Quality Improvement 21(10), 541–548 (1995)
4. Forster, A.J., Halil, R.B., Tierney, M.G.: Pharmacist surveillance of adverse drug events. American Journal of Health-System Pharmacy 61(14), 1466–1472 (2004)
5. Gandhi, T.K., Weingart, S.N., Borus, J., Seger, A.C., Peterson, J., Burdick, E., et al.: Adverse Drug Events in Ambulatory Care. The New England Journal of Medicine 348(16), 1556 (2003)
6. Gurwitz, J.H., Field, T.S., Avorn, J., McCormick, D., Jain, S., Eckler, M., et al.: Incidence and preventability of adverse drug events in nursing homes. American Journal of Medicine 109(2), 87–94 (2000)
7. ITU-T – International Telecommunications Union: Recommendation Z.151 (11/08): User Requirements Notation (URN) – Language definition. Geneva, Switzerland (2008)
8. Liu, L., Yu, E.: Designing Information Systems in Social Context: A Goal and Scenario Modelling Approach. Information Systems 29(2), 187–203 (2004)
9. Michel, P., Quenon, J.L., de Sarasqueta, A.M., Scemama, O.: Comparison of three methods for estimating rates of adverse events and rates of preventable adverse events in acute care hospitals. British Medical Journal 328(7433), 199–203 (2004)
10. Pourshahid, A., Chen, P., Amyot, D., Forster, A.J., Ghanavati, S., Peyton, L., Weiss, M.: Toward an integrated User Requirements Notation framework and tool for Business Process Management. In: 3rd Int. MCeTech Conference on eTechnologies, Montréal, Canada, January 3-15. IEEE Computer Society, Los Alamitos (2008)
11. Stead, W.W., Lin, H.S.: Computational Technology for Effective Health Care: Immediate Steps and Strategic Directions. In: Committee on Engaging the Computer Science Research Community in Health Care Informatics, National Research Council, USA. National Academies Press, Washington (2009)
12. The Institute of Medicine. To err is human: building a safer health system. National Academy Press, Washington D.C (2000)
13. Univ. of Ottawa: jUCMNav 3.2 (2008), `http://jucmnav.softwareengineering.ca/jucmnav/`
14. Weiss, M., Amyot, D.: Business Process Modeling with URN. International Journal of E-Business Research 1(3), 63–90 (2005)
15. van Lamsweerde, A.: Requirements engineering: From System Goals to UML Models to Software Specifications. John Wiley & Sons, Chichester (2009)