Towards the Security and Privacy Analysis of Patient Portals

Janos L. Mathe[1] Sean Duncavage[1] Jan Werner[1]

Bradley A. Malin[2] Akos Ledeczi[1] Janos Sztipanovits[1]

[1]Institute for Software Integrated Systems
Vanderbilt University
Nashville, TN 37209
{first}.{mi}.{last}@vanderbilt.edu

[2]Department of Biomedical Informatics
Vanderbilt University
Nashville, TN 37209
b.malin@vanderbilt.edu

ABSTRACT

Clinical information systems (CIS) significantly influence the quality and efficiency of health care delivery. However, CIS are complex environments that integrate information technologies, human stakeholders, and patient-specific data. Given the sensitivity of patient data, federal regulations require healthcare providers to adopt policy, as well as technology, protections for patient data. Ad hoc system design and implementation of CIS can cause unforeseen and unintended privacy and security breaches. The introduction of model-based design techniques combined with the development of high-level modeling abstractions and analysis methods provide a mechanism to investigate these concerns by conceptually simplifying CIS without losing expressive power. This work introduces the Model-based Design Environment for Clinical Information Systems (MODECIS) - a graphical design environment that assists CIS architects in formalizing CIS systems as well-defined services. MODECIS leverages Service-Oriented Architectures to create realistic system models at an abstract level. By modeling CIS using abstractions, we enable the analysis of legacy architectures, as well as the design and simulation of, future CIS. We present the feasibility of MODECIS via modeling certain functions, such as the authentication process of the MyHealth@Vanderbilt patient portal.

1. INTRODUCTION

Health care systems with errors that are difficult to detect and address can lead to serious mistakes in patient care. To reduce errors, many health-care organizations have migrated from paper-based to Electronic Medical Records (EMR), which have been shown to increase both staff productivity and patient safety [1]. Expanding on the success of EMRs, Clinical Information Systems (CIS) are part of an emerging technology that incorporates a wide range of the informational and organizational components of the health-care environment.

Local and federal regulations concerning the management of patient information influence CIS design and implementation. The Privacy Rule of the Health Insurance Portability and Accountability Act (HIPAA) specifically grants patients the right to access their medical records as well as request corrections and disclosures of their personal health information [2]. The HIPAA Security Rule additionally requires healthcare organizations to provide security protections at the physical, technical, and administrative levels to log access to identifiable health information [3]. Patient Portals are one method to accommodate the Privacy Rule and provide patients with a simple method to access their medical records, disclosures, and audits. Designing such a system optimally to protect patient confidentiality and respect health-care providers' rights is an open problem.

We begin to address this challenge by casting patient portals, a key portion of CIS, onto a Service-Oriented Architecture (SOA). We developed a domain-specific modeling environment called Model-based Design Environment for Clinical Information Systems (MODECIS) with which we create formal models of healthcare services and features for detailed analysis.

Our initial research with MODECIS successfully demonstrates that patient portals can be modeled as SOA. The development of critical modeling abstractions adds the feature of scalability to our tool. Although MODECIS is a work-in-progress, it is already able to express multiple aspects of patient portals and has been used to create high-fidelity models of the MyHealth@Vanderbilt patient portal, which relate to larger CIS operation.

2. BACKGROUND

Patient portals provide means to view, and contribute to, one's medical records; however, this patient integration also creates complex policy and technology management issues. Addressing these complications and adhering to both the HIPAA Privacy and Security Rule is a major concern for CIS.

Security was addressed in the PCASSO patient portal system, which, among other features, provided patients and physicians online access to medical records and the ability to audit these records [4]. Pilot studies reported positive effects on patient care while maintaining the security of patient records, evidenced by zero reported security breaches in the portal systems. Despite the apparent success of the pilot studies, applying the same design strategies to more complex systems with large numbers of users and available functions may not yield the same positive results.

As opposed to ad hoc design strategies, SOA is a web-inspired architectural style that enables extensible interoperability by using loosely coupled, interacting services to compose complex applications [5]. SOA calls upon independent, heterogeneous components, known as services, which can be accessed through predefined interfaces and composed into a workflow representing business logic [6][7]. The principal design goals of SOA services are composability, adaptability, and platform independence, which lead to improved interoperability among systems and future extensibility.

Workflows are a conceptual tool that can help capture the business logic of a system and are a cornerstone of the Business Process Execution Language (BPEL) [8]. BPEL is part of a group of SOA orchestration languages, which describe processes from a single point of view (such as the view of the patient portal in CIS), and it is complemented by a suite of standards for access control and security policy modeling.

Model Integrated Computing (MIC) is another design strategy that leverages models to capture the requirements, architecture, and the environment of system in high-level models [9]. The models can have multiple aspects to capture the actual structure of the system in design and the environment in which it will be deployed. For example, one can imagine models with aspects depicting the software components of a system (such as an http mirror), and the physical location of these components (such as the server hosting the mirror). The models can also act as a repository of information, capturing the necessary knowledge for analyzing and generating the system.

SOA has been previously proposed for the design of formally-composed CIS environments [10]. However, current implementations are limited by the fact they do not model patient-provider interactions. In this paper, we show how SOA can be applied to a specific patient-associated environment.

3. APPROACH

Workflows in BPEL (and in general) provide a representation of the manner by which data is accessed, handled, and shared. Without formal representations of daily business processes and their interrelationships within the healthcare environment, it is not clearly evident why a patient's medical record is accessed or how the interactions between patient and provider are managed. Both underspecified and ad hoc workflow design can lead to malformed policies with unanticipated consequences, and even seemingly routine business processes can lead to serious privacy compromises when taken in combination [15]. Taking this into account, formal workflow models are a starting point for the development and analysis of policy-driven operations supporting privacy and security.

This inspired the creation of the building of the tool suite, called MODECIS, where the formal basis of our approach allows for the extension, reuse, and evolution of clinical information systems [Figure 1].

MODECIS has three main components: a) a graphical design environment for capturing the business logic of CIS through workflows, b) an analysis tool, which allows for the analysis of information flows and the exploration of security and privacy properties of a CIS system modeled with the graphical design environment, and finally c) a model translator that maps the CIS-specific workflows to BPEL, WSDL and XACML. By translating the domain models onto these SOA standards, using the model transformation tool (GReAT) of the MIC tool suite [16], the underlying alternative implementations of SOA platforms for the standards become applicable. This radically simplifies the fast prototyping, integration and testing tasks.

By capturing the appropriate level of abstraction, it is possible to satisfy utility, security, and policy requirements for CIS. In MODECIS workflows – common in SOA – provide us with this abstraction layer, which is suitable for patient-centered clinical information representation and management. MODECIS with the workflow abstractions will allow us to perform vulnerability, security and privacy analyses through model verification and simulation-based testing tools. In addition to that; model-based design will provide the tools for automated system generation directly from the models.

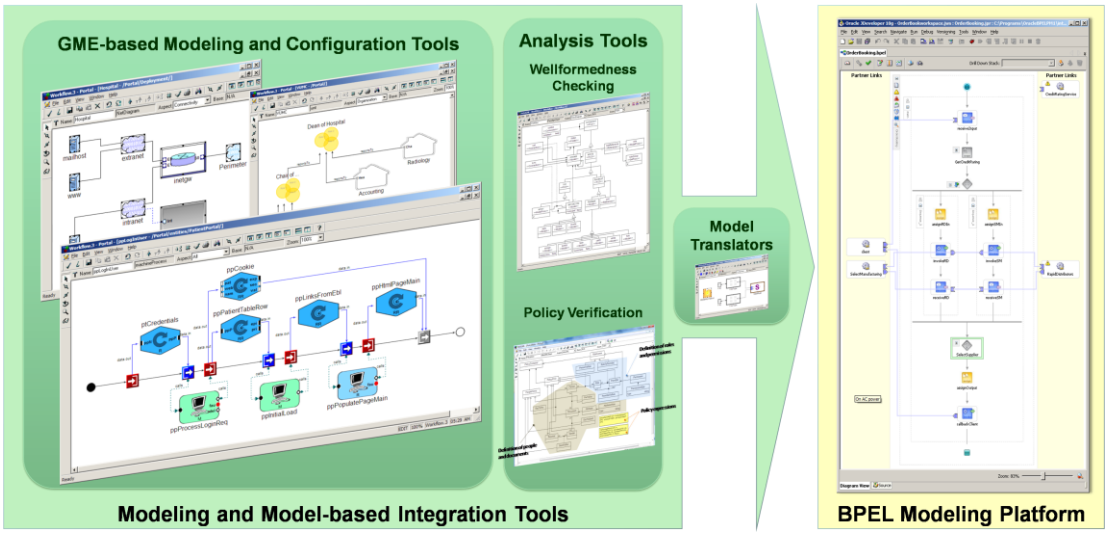

Figure 1 - MODECIS tool suite

3.1 Modeling abstractions

In MODECIS the graphical language for creating and managing workflows for CIS is based on BPEL, but it is customized to specifically capture the EMR context. We found the development of a domain-specific modeling language necessary after the unsuccessful efforts of trying to represent every aspect of a given CIS in BPEL. In MODECIS the domain-specific modeling language specifications are captured in the form of metamodels – (UML-based) models that define the language itself [GME ref]. Then, through an automatic generation process, the domain-specific design environment is synthesized and a new GME "instance" is created for the modeling of each CIS.

At the heart of our approach, the domain-specific modeling language captures the system from multiple viewpoints. And while the detailed description of this modeling language is beyond the scope of this paper, we do present a brief description of the modeling abstractions that help to capture the different aspects of the multi-facetted CIS domain. These modeling abstractions appear as different types of models in GME.

The *workflow models* can be thought of as a graphical equivalent of a simplified BPEL representation. They capture the orchestration logic with graphs that describe control, which specify the sequence of service invocations and data flows that represent the movement of information within a CIS system. The first aspect allows for the orchestration of control flows that are defined as a composition of service invocations – which can either be asynchronous or synchronous – and the typical control structures – such as switch, join, while, and catch – which allows for the definition of arbitrary workflow logic. The second aspect of workflow modeling describes the flow of data elements: how these elements are exchanged, processed and stored between and within various processes. This way each workflow model can be thought of as an available service with well-defined interfaces.

Since the workflow models only describe how data elements are used we have created the view for building *datatype models* (in a hierarchical fashion) which allows the language to be strongly typed.

Workflows in general allow system architects to follow the information traveling between entities and can represent diverse entities interacting with system, such as physical databases or people. For this reason MODECIS incorporates two more types of models for the integration of workflow models with the underlying architectures and physical entities. This means that a complicated, explicitly represented social and technical architecture can be constructed that the services build on.

The creation of *organizational models* allows for the human coordination within CIS. These models are used to specify the architecture of the enterprise itself, such as the roles of different people. Organizational models reflect inter- and intradepartmental interactions, as well as people's roles within departments specifying tasks and groups to whom these tasks are assigned. For example, they are referred to by policies to facilitate role-based access control.

While organizational models relate human-based workflow (i.e. workflows that describe expected behavior of and tasks preformed by the human players in CIS), *deployment models* specify the organization of computer servers, their conjunctive networks and interface with workflows in a similar manner to organizational models. They are often referred to as the network architecture (ex: they depict hospital servers and workstations along with the services they provide).

The final abstraction captures policy statements that crosscut workflow, organizational and deployment models. They place restrictions on accessing certain services and information. We have looked into modeling policies as a set of OCL expressions similarly to the method in [14].

3.2 Model analysis

The built-in constraint manager of GME is used for checking the models against structural violations. While we planning to either develop a suite of analysis tools for static model verification, the existing constraint checker already provides a powerful method to force modelers not to violate domain specific design rules (c.f. correct-by-construction).

As previously mentioned, MODECIS will include a model translator capable of mapping domain-specific models to executable BPEL code. Despite its wide acceptance, BPEL provides no support for the detection of a) possible deadlocks or b) process paths that are not viable. For the so-called workflow nets (a type of Petri nets), techniques and tools exist which make it possible to detect such anomalies. The idea proposed in [11] claims to resolve this problem by mapping BPEL process models onto workflow-nets. Existing research on modeling and verifying BPEL processes with the help of Petri Nets, SPIN model checker, Process Algebras, Abstract State Machines (ASM), Automata, etc. is nicely summarized in [12]. MODECIS plans to capitalize on these existing technologies for (BPEL) model verification.

One major advantage of using OCL for policy representation is that the MODECIS tool suite (specifically GME) has native support for OCL in the form of a parser and expression evaluator. We leverage this asset for static policy design and enforcement in the CIS domain.

The distribution of portal services across deployments raises complex logistical, privacy, and security concerns that we are planning to address with the analysis techniques mentioned above.

3.3 Execution engine

As a final, system integration step to guarantee correct flow of logic captured by the domain models, the tool suite interfaces with an execution engine, which after deployment manages the multiple instances of workflows. Specifically, the engine organizes and executes the services required by the CIS entities (e.g., a patient, primary care provider, and patient portal) and enforces policies.

We are currently using the Oracle BPEL Process Manager as our execution engine [13].

4. DISCUSSION AND CONCLUSIONS

The MODECIS tool suite provides a domain-specific, graphical design environment for precisely describing organizational, deployment, service, and data models in relation to patient portals.

Through our collaboration with Vanderbilt University Medical Center (VUMC), we were able to create a modeling language capable of representing a functional patient portal. The VUMC group was also able to confirm the expressiveness and correctness of our patient portal workflow models, which we have begun to deploy on the Oracle BPEL execution engine.

Although MODECIS is a work-in-progress, models created with the tool suite serve as formal system specifications that can be mapped onto various SOA execution platforms for simulation. Consistency and wellformedness checking is already supported by MODECIS; support for policy verification and vulnerability and security analysis of the models is our next step, which will be supported through the use of existing analysis tools.

MODECIS provides a scalable tool to evaluate design decisions and system changes before deploying costly healthcare infrastructure. The creation of patient portal models and simulations is one step toward designing robust CIS that are able to take into account the diverse privacy and security concerns of stakeholders.

5. ACKNOWLEDGMENTS

This research was funded in part by the Team for Research in Ubiquitous Secure Technologies (TRUST) NSF CCF-0424422, an NSF S&T Center. The authors wish to thank John Doulis, Dario Giuse, Jim Jirjis, Jun Kunavat, Dan Masys, Sue Muse, Bill Stead, Yun Wang, and especially Jim Weaver for the insightful discussions and their time. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government or any of its agencies.

6. REFERENCES

[1] Davies NM. Healthcare Information and Management Systems Society: The ROI of EMR-EHR: Productivity Soars, Hospitals Save Time and, Yes, Money. *HIMSS Journal*. 2006.

[2] U.S. Department of Health and Human Services. Standards for privacy of individually identifiable health information; Final Rule. *Federal Register*, 2002 Aug 12; 45 CFR: Parts 160-164.

[3] U.S. Department of Health and Human Services, Office for Civil Rights. Standards for protection of electronic health information; Final Rule. *Federal Register*, 2003 Feb 20; 45 CFR: Pt. 164.

[4] Masys D, Baker D, Butros A, Cowles KE. Giving patients access to their medical records: the PCASSO experience, *J Am Med Inform Assoc*. 2002; 9(2): 181- 91.

[5] A. Yanchuk, A. Ivanyukovich, M. Marchese: "Towards a Mathematical Foundation for Service-Oriented Applications Design", http://www.science.unitn.it/~marchese/pdf/Towards_SOAD_JoS_06.pdf

[6] B. Portier: "SOA terminology overview, Part 1: Service, architecture, governance, and business terms", http://www-128.ibm.com/developerworks/library/ws-soa-term1/index.html

[7] B. Portier: "SOA terminology overview, Part 2: Development processes, models, and assets", http://www-128.ibm.com/developerworks/library/ws-soa-term2/index.html

[8] OASIS: "Web Services Business Process Execution Language (WSBPEL) TC", http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

[9] G. Karsai, J. Sztipanovits, A. Ledeczi, and T. Bapty, "Model-integrated development of embedded software," *Proceedings of the IEEE*, vol. 91, no. 1, pp. 145-164, Jan. 2003.

[10] Kawamoto K, Lobach D. Proposal for fulfilling strategic objectives of the U.S. roadmap for national action on decision support through a service-oriented architecture leveraging HL7 services. *J Am Med Inform Assoc*. 2007; 14: 146-55.

[11] R. Hamadi, B. Benatallah: "A Petri Net-based Model for Web Service Composition", http://crpit.com/confpapers/CRPITV17Hamadi.pdf

[12] F. van Breugel, M. Koshkina: "Models and Verification of BPEL", http://www.cse.yorku.ca/~franck/research/drafts/tutorial.pdf

[13] Oracle BPEL Process Manager, http://www.oracle.com/technology/products/ias/bpel/index.html

[14] M. Alam, R. Breu, M. Hafner, "Modeling permissions in a (U/X)ML world," in *Proc. First International Conference on Availability, Reliability and Security*, pp. 685-692, April 2006.

[15] B. Malin and L. Sweeney, "How not to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems," *Journal of Biomedical Informatics*, vol. 37, no. 3, pp. 179-192, Feb 2004.

[16] G. Karsai, A. Agarwal., F. Shi, and J. Sprinkle, "On the use of graph transformation in the formal specification of model interpreters," *Journal of Universal Computer Science*, vol. 9, no. 11, pp. 1296-1321, Nov 2003.