A Modeling Environment for Patient Portals

Sean Duncavage[1], Janos Mathe[1], Jan Werner[1], Bradley A. Malin[1,2], PhD, Akos Ledeczi[1], PhD, and Janos Sztipanovits[1], PhD

[1]Department of Electrical Engineering and Computer Science, School of Engineering
[2]Department of Biomedical Informatics, School of Medicine
Vanderbilt University, Nashville, TN

Clinical Information Systems (*CIS*) *are complex environments that integrate information technologies, humans, and patient data. Given the sensitivity of patient data, federal regulations require health care providers to define privacy and security policies and to deploy enforcement technologies. 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 sacrificing expressive power. This work introduces the Model-based Design Environment for Clinical Information Systems* (*MODECIS*)*, which is a graphical design environment that assists CIS architects in formalizing systems and services. MODECIS leverages Service-Oriented Architectures to create realistic system models as abstractions. MODECIS enables the analysis of legacy architectures and the design and simulation of future CIS. We present the feasibility of MODECIS by modeling operations, such as user authentication, of MyHealth@Vanderbilt, a real world patient portal in use at the Vanderbilt University Medical Center.*

INTRODUCTION

A health care system with errors that are difficult to find and correct can lead to serious mistakes in the treatment of patients. To mitigate errors, many health care organizations have moved paper-based records to Electronic Medical Records (EMRs), which have been shown to increase staff productivity and patient safety.[1] Beyond EMRs, Clinical Information Systems (CIS) are emerging as a technology that builds upon EMRs as a foundation and integrates a wider-range of the information and organizational components in a health care environment. Given the potential for errors and the sensitivity of patient information, the design of CIS is a critical issue that affects the well-being of patients.

Local and federal policies regarding the management of patient information influence the design and implementation of CIS. Specifically, the Privacy Rule of the Health Insurance Portability and Accountability Act (HIPAA) provides patients the right to access their medical records, request corrections to those records, and request a log of disclosures of their personal health information.[2] The HIPAA Security Rule requires health care organizations to implement security protections at the physical, technical, and administrative levels in order to monitor and document access to identifiable health information.[3] Patient portals are one way to meet the mandates of the Privacy and Security Rules and provide patients with a cost-effective method to view their medical records, disclosures, and access audits. However, the optimal design of such a system to protect patient confidentiality and respect the rights of health care providers is an unresolved challenge.

In this paper, we begin to address this challenge by casting patient portals, a 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), which we use to create formal models of system services and features for rigorous analysis. We seek to investigate the privacy and security implications of patient portals, determine the information passed among care providers and patients, and characterize where patient data is stored.

Our research shows that patient portals can be represented as a SOA through the application of MODECIS. In addition, our development of modeling abstractions for organizational and technical system components supports the scalability and reusability of our tool. MODECIS is a work-in-progress, but this work demonstrates that it is already capable of capturing multiple aspects of patient portals in the context of complex modern CIS.

BACKGROUND

Web portal technologies provide patients with the opportunity to view, and contribute to, their medical records. Since the mid-1990s, the National Institutes of Health has supported the development and prototyping of telemedicine systems for patient access to electronic medical records.[4-7] For example, the PCASSO system provided patients and physicians with online access to medical records, the ability to audit access to records, messaging between patients and physicians, and authorization for health care functions.[8] The users of these portal systems reported positive effects on care.

At the same time, the integration of patients into the realm of electronic medical records creates complex policy and technology management issues. Privacy and confidentiality of medical information is a major concern for CIS and is a key issue to address for successful deployment of such systems.[9] Computer security has been addressed in specific CIS and patient portals.[10-12] These protections were successfully applied in the aforementioned studies and no security breaches were reported in the portal systems. Yet, the pilot studies enrolled relatively small numbers of patients and were limited in their functionality. These systems may not be predictive of the societal effects of much broader availability of online medical records.

Medical systems have evolved from groups of medical portals, such as separate physician portals and patient portals,[13] into orchestrated applications that facilitate standardized communication among web-services.[14] Recently, SOA were proposed for the formal design of such medical systems[15] to aid clinical decision support systems and integrate standards, such as HL7. However, existing implementations are limited in that they do not model CIS in the context of patient-provider interactions. In this paper, we illustrate how SOA can be applied to a specific patient-centric environment.

METHODS

MyHealth@Vanderbilt Patient Portal

We evaluate the technical capability of MODECIS against the realities of patient care. We build on unique insight provided by clinical operations of the MyHealth@Vanderbilt (MHAV) patient portal, which is in operation at the Vanderbilt University Medical Center (VUMC). The MHAV portal is one of the more advanced online patient-centered health care sites in operation.[16] It provides a growing set of individualized web-based services to more than 25,000 enrolled individuals and approximately 50,000 care providers.

Services for patients include secure message-based communications between providers and patients that are automatically incorporated into the relevant electronic medical records, the ability for patients to request appointments online, the opportunity view the current status of their billing information, and access to personalized literature relevant to their health status. The MHAV website also offers the availability of clinical laboratory results and other components of the electronic medical record to patients that were previously available only to physicians and other health care providers.

To apply MODECIS to the MHAV portal, we held collaborative and iterative design sessions with system developers and administrators from the VUMC, where the patient portal was developed and is maintained. This collaboration enabled us to evaluate MODECIS in a real-world environment, which is crucial to demonstrate the effectiveness of the services-oriented design approach.

MODECIS and SOA

Here, we provide details into MODECIS and patient portal systems design.

MODECIS is build using the Generic Modeling Environment (GME), which is part of the Model Integrated Computing tool suite.[17-20] In GME, we defined a domain-specific modeling language that was designed for CIS environments. The architecture of MODECIS was based on the Business Process Execution Language for Web Services (BPEL4WS), which is a SOA standard designed for the orchestration of web services.[21] The MODECIS language is suitable for modeling organizational, workflow, service, and policy models of patient portals.

The MHAV portal is a component of a larger CIS at the VUMC where it interacts with the electronic medical record system. Our design meetings revealed that the system components of the VUMC CIS handle requests for accessing information and services through complex information flows that are constantly evolving. The VUMC CIS, and similar legacy systems, tend to consist of hard-wired information flows that obscure the orchestration logic. To make the logic more explicit, MODECIS leverages SOA to separate the low-level implementation details from the high-level abstractions. By recasting CIS into the SOA environment of MODECIS, we can generate a highly analyzable and dynamically reconfigurable orchestration layer that supports service interactions. This is possible because SOA was specifically designed for integration within complex enterprise systems to support business processes and software users.

MODECIS captures CIS orchestration logic with the assistance of workflow models. In this context, workflow models are directed graphs that describe control and data flows within the CIS system. It is crucial to model the workflows of CIS because they provide a representation of how patient data is accessed, handled, and shared. In addition, workflows allow system architects to characterize the interactions between diverse entities within the system, such as physical databases and people, as well as the information traveling between such entities. For a complete CIS model, we must integrate workflow models with the underlying architectures and physical entities. This means that there is a complicated social and technical architecture behind the services of the MHAV patient portal, which must be explicitly represented.

By capturing the appropriate level of abstraction, MODECIS makes it possible to fulfill utility, security, and policy requirements for CIS. For example, workflow abstractions allow system architects to perform vulnerability, security and privacy investigations through enterprise-level model verification and simulation-based testing tools. SOA provides abstractions for patient-centered clinical information management. In contrast, the model-based design and system integration task focuses on the formal representation of information processes and security/safety policies. In addition, model-based design provides the tools for automated system generation directly from the models.

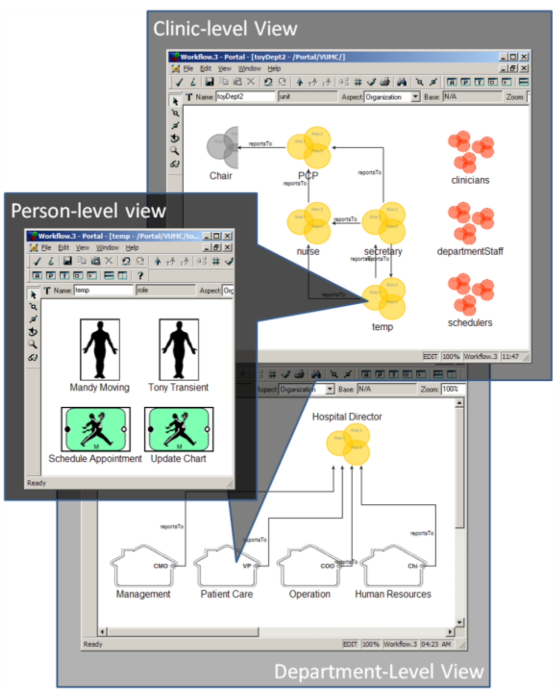

Fig 1. Organizational models of roles and groups in the MODECIS environment.

RESULTS

To demonstrate and evaluate the expressiveness of MODECIS, we constructed models of the MHAV patient portal. Collaboration with portal developers provided a key asset in creating and validating our representations. The resulting, high-fidelity models focus on four views of CIS: organization, deployment, service, and data.

Organizational Abstractions

To capture the human coordination within CIS, we applied MODECIS to create organizational models (Figure 1) that reflecting inter- and intradepartmental interaction, as well as roles within departments. Interdepartmental interactions capture the communication of information between separate clinical entities (e.g., hardware servers and human service providers), such as what would occur when a patient is referred by a health-care provider to a specialist in another department. In contrast, intradepartmental interactions express the flow of information within a single clinical department. For example, an individual who schedules appointments within the gastroenterology department can have a different role, title, and set of responsibilities than an individual who performs the same task in the internal medicine department. This is particularly so when temporary employees, who move between departments and responsibilities, are taken into consideration. Acknowledging this, the MODECIS models contain roles specifying tasks and groups to whom these tasks are assigned.

Deployment Abstractions

The distribution of portal services across deployments raises complex logistical, privacy, and security concerns. We can address these issues through deployment models that capture the coordination of machines within CIS. This is often referred to as the network architecture. MODECIS models of MHAV depict hospital servers and workstations along with the services they provide. For instance, the MHAV server is housed separately from the hospital EMR server, but both provide portal services.

Service Abstractions

The MHAV service models define the workflows of hospital staff and portal-related software. Each workflow is composed of control flows (pictured at the top-right of Figure 2), which specify the sequence of service invocations, and data flows (pictured at the bottom-right of Figure 2), which specify the movement of information.

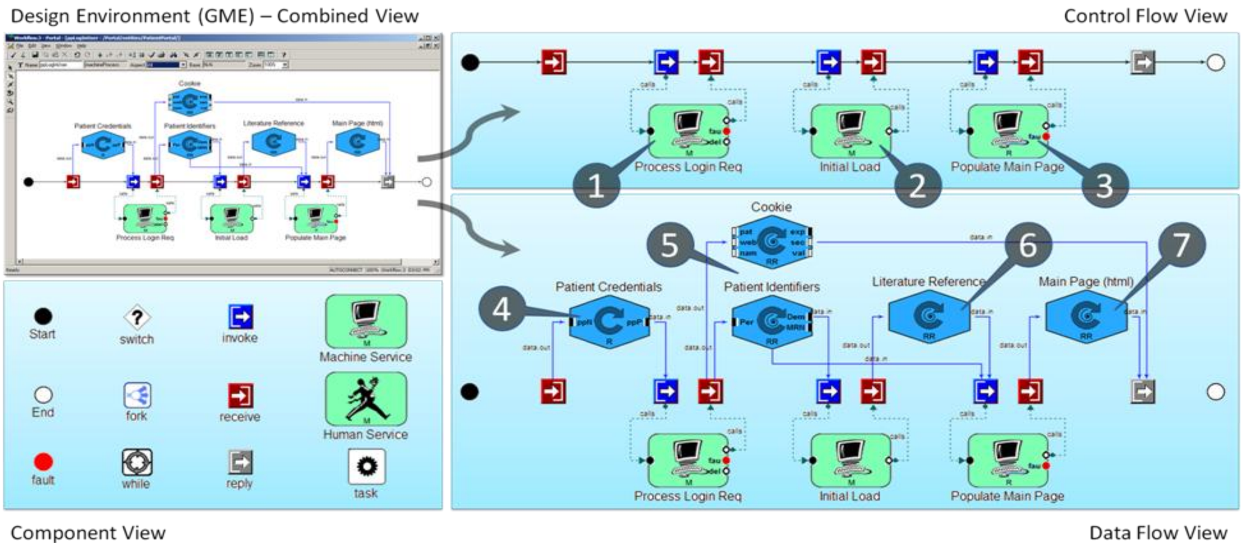

Fig 2. Workflow models of MyHealth@Vanderbilt derived through MODECIS.

Data Abstractions

Structured information in MHAV is defined in our data models (the blue hexagons in Figure 2). These define the data primitives required by portal services, such as the patient's MRN or the current time. The data models can be either complex or simple types: For example, we use a simple type to represent an integer MRN and a compound type to represent a web address containing a MRN and service calls.

MHAV Workflow Example

The following example depicts the level of abstraction MHAV stakeholders and system architects found necessary to express the components, processes, and logic of MHAV. The workflow in Figure 2 walks through the series of data transactions and service invocations that occur when a user logs into MHAV. Though login is often considered a trivial issue, the workflow illustrates that, even at the time of login, a patient's protected health information can be accessed.

The control flow (Figure 2, Control Flow View) initiates when a patient logs onto MHAV. This is represented by the black dot, which indicates that the system waits for some information before proceeding. The waiting process is represented by the receive arrow in the Component View of Figure 2. The invoke arrows serve as starting points for new workflows. The control flow, represented by the black line, shows the sequence of events that occur before the patient sees a customized home page.

Following the control flow, there are three service calls that are made from the portal: 1) a service to retrieve patient information; 2) a service that retrieves additional patient information; and 3) a service to assemble these pieces into the patient's personalized MHAV home page. Upon the completion of the service calls, the resulting page is returned to the user. Finally, the workflow ends at the white dot.

Inspection of the Data Flow in Figure 2 provides another view of the system that focuses on the data inputs and outputs of invoked services. In this view, it is specified that before a patient's login request can be processed, the patient must have provided authorized credentials in a previous workflow. 4) Once the credentials have been provided, 5) they are processed to generate a cookie for the patient's web browser and to retrieve patient identifiers for further service invocations (such as the unique patient ID and MRN). 6) After the login request is processed, a patient identifier is used to retrieve reading materials that are relevant to the patient's health conditions. 7) A final service generates the page seen by the patient.

The level of granularity in the workflow provides a clear separation of system processes and enables system architects to track the movement of protected health information through a CIS. Consequently, we have the ability to simulate privacy policies and access control techniques. The abstraction of health care services in this way also assists in the maintenance and distribution of services, which can be on geographically distant servers, because it correlates entities with the services they provide.

DISCUSSION AND CONCLUSIONS

Medical information is increasingly accessible in health care and online environments. It is necessary to understand the impact of such a shift in access on both patients and health care organizations. The creation of patient portal models and simulations is one step toward designing scalable, robust CIS that explicitly accounts for the stakeholders' diverse privacy and security concerns. MODECIS provides a domain-specific, graphical design environment for precisely describing organizational, deployment, service, and data models. MODECIS's formal system specifications can be mapped onto various SOA execution platforms for simulation and allows us to perform consistency and well-formedness checking, policy verification, and vulnerability analysis. As next steps, we intend to investigate alternative patient portal implementations in SOA and their effects on various privacy policies and security measures.

Acknowledgements

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.

References

1. Davies NM. Health care 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. Baker D, Barnhart R, Buss T. PCASSO: Applying and extending state-of-the-art security in the health care domain. *Proc ACSAC.* 1997: 251.
5. Cimino JJ, Patel, VL, Kushniruk AW. The patient clinical information system (PatCIS). *Int J Med Inform*. 2002; 68(1-3): 113-27.
6. Masys D, Baker D. Patient-centered access to secure systems online (PCASSO): a secure approach to clinical data access via the world wide web. *J Am Med Inform Assoc*. 1997; 4: 340-3.
7. 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.
8. Masys D, Baker D. Protecting clinical data on web client computers: the PCASSO Approach. *Proc AMIA Symp*. 1998: 366 -70.
9. Barrows RC Clayton PD. Privacy, confidentiality and electronic medical records. *J Am Med Inform Assoc*. 1996; 3(2): 139-48.
10. Baker DB, Masys, DR. PCASSO: a design for secure communication of personal health information via the internet. *Int J Med Inform*. 1999: 5: 97-104.
11. Masys DR, Baker DB, Barnhart R, Buss T. PCASSO: A secure architecture for access to clinical data via the Internet. *Proc. MEDINFO*. 1998; 9 Pt 2:1130-4.
12. Ferreira A, Correia R, Costa-Pereira A. Securing a web-based EPR: an approach to secure a centralized EPR within a hospital. *Proc 6th International Conference on Enterprise Information Systems*. 2004: 54-9.
13. Shepherd M, Zitner D, Watters C. Medical portals: web-based access to medical information. *Proc 33rd HICSS*. 2000: 5003.
14. Anzbock R, Dustdar S. Modeling and implementing medical web services. *Data and Knowledge Engineering*. 2005; 5(2): 203-36.
15. 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.
16. https://www.myhealthatvanderbilt.com/
17. Karsai G, Nordstrom G, Ledeczi A, Sztipanovits J. Specifying graphical modeling systems using constraint-based metamodels. *IEEE Symp on Computer Aided Control System Design*. 2000: 89-94.
18. Sztipanovits J, Karsai G. Model-integrated computing. *IEEE Computer*. 1997; 30: 110-2.
19. Balasubramanian K, Gokhale A, Karsai G, Sztipanovits J, Neema S. Developing applications using model-driven design environments. *IEEE Computer.* 2006; 39: 33-40.
20. Ledeczi A, Bakay A, Maroti M., Volgyesi P, Nordstrom G, Sprinkle J, Karsai G. Composing domain-specific design environments. *Computer*. 2001; 34(11): 44-51
21. OASIS Web Services Business Process Execution Language. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel