Modeling Privacy Aware Health Information Exchange Systems

Andras Nadas[1], Mark E. Frisse[2] and Janos Sztipanovits[1]

[1] Institute for Software Integrated Systems at Vanderbilt University, Nashville, Tennessee, USA
andras.nadas@vanderbilt.edu, janos.sztipanovits@vanderbilt.edu
[2] Vanderbilt University Medical Center, Nashville, Tennessee, USA mark.frisse@vanderbilt.edu

Abstract

Health care delivery requires the coordination of activities across many different service providers and organizations and generally requires the secure exchange of health information across organizations. Privacy and care management policies at the federal, state, and institutional level lead to a confusing array of potentially contradictory or subjective policy interpretations. Enforcement of these policies is equally problematic. In this paper we present a novel approach that may simplify policy creation, interpretation, and implementation in health information systems. Our approach uses the tools of Model Integrated Computing and Logic Programming to iteratively reduce the gap between functional system requirements and the privacy policies that are mandated by the Government and the Institution. This approach will provide both insight and explanatory power that may lead to simpler and more consistent policies across jurisdictions. The automatic generation of executable code from the system suggests a scalable approach to policy enforcement.

1 Introduction

Health care delivery requires the coordination of activities across many different service providers and organizations. In the United States, elderly patients covered under the Medicare Program see a median of seven different physicians [27]. Providers caring for a patient must have all relevant information available when they make decisions and plan care. Seldom is this information available in a single electronic record system. In the United States, the introduction of interconnectivity standards has been accelerated both by the realization of the potential for electronic medical records (EMRs) to provide better care and by the availability of large Federal financial incentives [7] [19]. The current challenge in using health information technology to improve care is no longer the availability of EMRs in practice systems but instead the ability of these systems to exchange information to coordinate and improve care.

As is the case in every other area of digital life, privacy is a major concern. Confidentiality of health information is a major public concern. Over the past two decades, numerous steps have been taken to create policies that govern the use and exchange of health information. At the federal level, the Health Insurance Portability and Accountability Act (HIPAA) introduced in 1996 set new standards for use of health information [32]. Standards were strengthened with the introduction of the Health Information Technology for Economical and Clinical Health (HITECH) act of 2010. This act, and subsequent federal rules, have mandated harsh financial penalties for the inappropriate disclosure or use of personal health information [33]. In response to public concern and the introduction of digital health information technologies, state governments as well have revised may of the laws to impose constraints and penalties on the use of personal health information. States differ widely in how they define specific concepts (e.g., a minor, psychiatric information) and at times are ambiguous as to the precise meaning of terms (e.g., qualified providers).

Facing these same concerns, hospitals and large ambulatory care organizations have modified their data use policies to provide their own practices for health information use. Although there are some rules for precedence, ensuring consistent harmonization of policies across federal, state, and institutional policies remains a major challenge [8].

In the European Union (EU) the issue of patient privacy is approached from the opposite direction as in the United States (US). In the EU the protection of privacy is considered stronger than in the US and the individuals are granted with more control over their privacy [10].

A secure and consistent means of implementing policies both for access and for care coordination will be essential. To realize these policies, people, process, and technology must be integrated to accommodate an ever-shifting set of clinical requirements, laws, policies, and patient preferences. Responding to a request for information or other specific actions will depend upon the authority and identity of the requesting agent, the privacy rights of the patient, the nature of the data, the policies of the institution, and both state and federal laws. Ensuring consistent action within a single institution in a single state is a challenge; ensuring consistent action among institutions and individuals governed by different institutional policies and state laws is even more difficult.

Scalable and sustainable health information exchange (HIE) among EMRs will require flexible systems capable of modeling ever-changing policies and complex concepts that differ among individuals, organizations and states. Rigid systems may enforce policies in the here and now, but the complexity of the privacy domain makes frequent modification and widespread adoption inconsistent and error-prone.

As the Model Integrated Computing (MIC) methodologies are becoming mature it becomes possible to fuse multiple design methods together and leverage their power to solve interdisciplinary problems. Model Integrated Computing is a design paradigm where the system under design is modeled from many different aspects to provide a full description. Such aspects can include an actual architecture, requirements or the environment of the system. These models enable analysis, transformation and reasoning on the system they model using external tools. Although the MIC paradigm requires an extensive suite of tools to achieve this, such tools are becoming more available. One comprehensive tool suite is the Generic Modeling Environment (GME) [17] that includes a graphical modeling user interface with the Universal Data Model (UDM) package and the Graph Rewriting and Transformation (GReAT) package [14]. The GME toolsuite includes support for using FORMULA (Formal Modeling Using Logic Programming and Analysis) from Microsoft Research as an analysis framework [13] [12] [11]. The GME tool suite has a proven record in providing modeling for health system [18] and other mission critical systems [28].

2 Related Work

With the push for more open health information systems the question of privacy became a high priority problem and a very active research area. The theory of privacy for computer systems build on the framework of Contextual Integrity [26] that establish the term privacy using contexts where the same social norms are expected. Contextual Integrity can be used as a framework to formalize and reason about the norms of transmission of personal information [4] in general as well as in medical systems [16]. Once formally represented the privacy policies can be used to verify system behavior and to provide automated approach for enforcement [3] and audit [6]. The presented modeling approach builds on these results to provide a higher level of abstraction that enables the involvement of stakeholders outside the Information Technology departments.

All privacy frameworks build on the security layer of the system for encryption and authentication. Recent research results of encryption enable new means for scalable and straightforward implementation of disclosures using Attribute Based Encryption [15]. Using attributes with contexts also enables the creation of flexible and more comprehensive policies for authentication to control the access to different health information systems [21]. Controlling the access to databases containing sensitive medical information while maintaining privacy of the patient is crucial [1]. Health care database systems that take privacy into consideration are called Hippocratic Databases [2] after the Hippocratic oath that Physicians make to practice medicine ethically. Hippocratic databases can be extended to act as simple HIE systems for disclosures while maintaining a consistent set of privacy restrictions [30]. Health care Information Systems including HIE systems can be treated and designed as Federated Healthcare Database that takes privacy policies into consideration [5]. All the above-mentioned approaches uses policies to control the access to and operation of the Health Information Systems or the disclosure of information. The modeling approach presented in this paper can be used to author, verify and provide these policies.

Research into engineering approaches very similar to our efforts was conducted in the areas of information flow modeling and policy based testing. Information flows represent the core concept in HIE systems and as such has to be modeled with utmost care taking into account the flows might be going through untrusted systems. A decentralized solution using distributed labeling [23] has been shown to give scalable results while protecting privacy [24] . Once a policy-based system is designed and developed the next phase is testing. The common incremental test methodologies might fail on these systems as a small change in policies can have major change in the systems behavior. One solution to this problem is to test the system using use case scenarios that describe the systems behavior [31].

3 The PATRN MIC toolkit

The Policy Authoring and Reasoning (PATRN) toolkit is an MIC tool suite built upon GME [25]. It provides modeling tools to formalize and reason about privacy policies and this functionality is where its name originates. The core functionality of the toolkit is to provide means to model the privacy policies with the associated ontologies and to compose them with formal semantics. The toolkit also provides modeling tools to describe systems and the information flows between them. The system and information flow models can be used exercise the policy models and test how the policies affect the systems behavior.

The Modeling process using the PATRN toolkit happens as shown in Figure 1. The first step is to recognize the common patterns, objects and actors in the federal and state level as well as the institution level textual policy descriptions. The next step is to compile the object and actors of the policies and organize them into ontologies. Once the patterns and ontologies are defined, policy models can be composed from them. The semantics of the relations and patterns are best defined by using a logic programming language but the PATRN toolkit provides a flexible code template framework that can be used with other programming languages as well. The system models and the actors of the systems are modeled based on the architectural and requirement specification of the HIE system. The use case models can be created based on the usage scenarios of the system in design. Once the modeling is complete the models can be exported and used in different contexts. The two contexts shown on Figure 1 are Verification and Enforcement. The toolkit enables the models to be composed with different semantics and exported into different formats. For example for use case simulation and verification the models can be anchored to semantics defined by Formula and analyzed by the Formula execution engine.

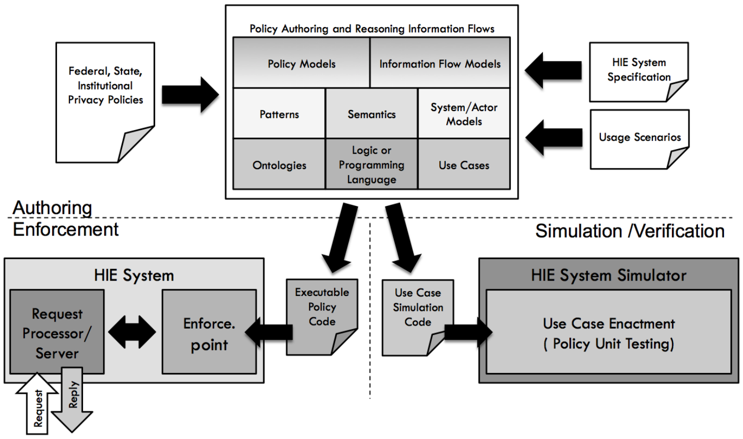

Figure 1: The components of the PATRN framework.

On the other hand the policies can be anchored to DROOLS rule engine semantic and used as rules in a policy enforcement point.

The ontology language in PATRN is a graphical derivative of the Ontology Web Language (OWL) [20] that is specified by the meta-model of PATRN. The graphical language is capable of describing classes, individuals, inheritance relationships, and properties with the same semantics as specified by OWL. The adaption of the semantics of OWL to the PATRN toolkit enables the integration, import and export of ontologies from already existing sources and systems.

The patterns provide a reusable structure with structural semantics enforced upon specialization. Patterns can be viewed as an extension of the OWL properties with structural semantics, and represent an n-ary association between individuals of classes. The patterns themselves do not imply or possess any operational or denominational semantics. These are assigned to the patterns by anchoring their semantics with a formal specification that matches their intended final use. The formal specification is provided using logic program fragments and templates. These fragments and templates are completed automatically when the patterns are specialized to crate the policy models. This separation of the structural and behavioral semantics enables the use of the same patterns and their instantiated models in different target domains, such us analysis, verification and execution. Patterns can represent not just the policy language patterns but also relations between individuals and constraints too.

The actual information flows are specified by sequences of episodes. Each episode describes a set of transactions where new documents are being pushed into or retrieved from one of the systems. This information flow representation is a high level specification and is designed to be able to represent a desired workflow over the systems during a treatment of single patient. This high level approach grants a great level of flexibility of the models while providing an easy understanding for stakeholders that are not necessarily technical.

4 System and Policy co-modeling approach using PATRN

To model the privacy policies together with the system an approach containing five cycles can be used. In each cycle of refinement the abstraction level and the design gap is decreased as show on Figure 2. The five iterations are the following. The first step is specifying and

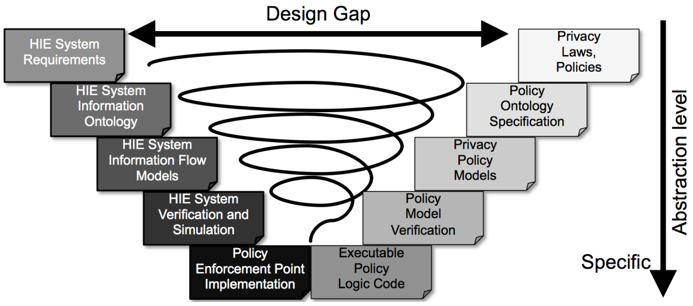
Design Gap
HIE System Requirements
HIE System Information Ontology
HIE System Information Flow Models
HIE System Verification and Simulation
Policy Enforcement Point Implementation
Privacy Laws, Policies
Policy Ontology Specification
Privacy Policy Models
Policy Model Verification
Executable Policy Logic Code
Abstraction level
Specific

Figure 2: The iterative System and Policy co-design and how it reduces design gap.

collecting the system requirements and collecting the policies that will govern the system. In the second iteration the information stored in the system and its representation is modeled using ontologies. Also in the second step the concepts and patterns of the policies are extracted and modeled. The third iteration step is to model the system components and information flows of the system as well as modeling and formally specifying the policies. The fourth iteration step is to compose the policy and system models by mapping the concepts of the policy ontologies to system ontologies. Also in this iteration the policy models are verified both for internal consistencies and composed with the system models. The last iteration step is to configure and implement the system and to translate the verified policy models into executable logic code. In the next sections each of these iterations are presented in detail together with a use case is that is used to demonstrate the approach.

4.1 A simple use case

General Hospital is general practice hospital and has an older health information system consisting of separate EMR, Computerized Physician Order Entry (CPOE), Admissions and Patient Portal systems, all from different vendors. An external system integrator did the integration of all these systems. Star Hospital on the other hand is very reputable specialty hospital, where General Hospital frequently refers patients for specialty care. Star hospital has a newer health information system that covers all the necessary functions that is provided by single vendor. However for data safety reasons the patient portal system is still separate from the rest of the system.

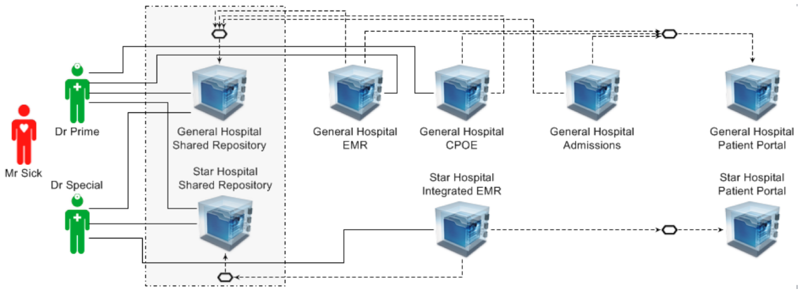
Mr Sick
Dr Prime
Dr Special
General Hospital Shared Repository
Star Hospital Shared Repository
General Hospital EMR
General Hospital CPOE
General Hospital Admissions
General Hospital Patient Portal
Star Hospital Integrated EMR
Star Hospital Patient Portal

Figure 3: The current system architecture of the two hospitals of the use-case showing the planned expansion in the light box.

To improve the care they deliver and to lessen their administrative burden General Hospital and Star Hospital are in talks to create a data sharing agreement and create systems supporting that. Under the data sharing agreement the two hospitals agree on creating a separate system where they replicate the data of the patients that are under the care of both institutions and grant access to physicians caring for those patients. It has to be noted that the same modeling approach can be used in cases when there is no common system existing and established between the different participants. In this case result of the modeling will be a minimally necessary specification of structures and policies to establish some sort of connectivity between the participants.

To exercise the planned systems the following use case was created. This use case covers the everyday workflows of referring patients between the two hospitals. Mr. Sick usually visits General Hospital as a patient. Dr. Prime is an internist at General Hospital and the primary care physician (PCP) of Mr. Sick. Dr. Special is a surgeon at Star Hospital. At one the regular checkups Dr. Prime notices an abscess on Mr. Sick. He decides to refer Mr. Sick to Star Hospital for a surgery. He documents this referral. When Mr. Sick shows up at Star Hospital for the surgery Dr. Special is assigned to his care. Before the surgery Dr. Special needs to check the medical history of Mr. Sick. After the surgery Dr. Special documents the outcome of the procedure. Next time Mr. Sick visits Dr. Prime for his regular check up Dr. Prime will check on the discharge notes form surgery.

4.2 Iteration 1: Requirements and Policy specification

The first step is to collect and create all requirements specification of the HIE system being designed. This includes the architecture and functional documentation of all currently existing system components, as most of the new HIE systems are built to extend already existing health information systems. This documentation has to specify the types and semantics of data available in the system. The architecture and functional requirements of the HIE system being designed including what data will be and can be shared between the participants. The list and text of all the policies, including the privacy policies, that will have an effect on the system behavior. These policies include the Federal policies and laws, the state and local government specified policies, the institutional policies and workflows, as well as requirements extracted from the data sharing agreements.

The system architecture and functional documentation bundle will serve as basis of all the system and information flow models. The precision and detail of the models depend on the quality of the documentation collected here. The system architecture and the data models will be converted into the system models containing the document models. The functional specification and workflows will serve the base for the information flow models.

The privacy and access policy models will be created based on the data sharing agreement, the institutional polices and the federal and local government mandated policies. At this point the designers have to make a decision about which policies are going to be treated as static or dynamic constraints on the system. The policies that are static constraints on the systems can be merged into the system requirements and implemented using the standard methods. The dynamic polices are going to be explicitly enforced at run-time, or will be audited. Some of these policies may fall outside the set for which automatic processing is feasible. These policies have to be manually checked and audited.

The policies that will actually affect the sample use case, are the HIPAA policies especially ones regarding Opt-out, local policies controlling the disclosure of sensitive medical information for example psychiatric notes and substance abuse records, policies from the data sharing agreement which limits the access to the shared data only to treating physicians.

4.3 Iteration 2: Ontology extraction and modeling

The second iteration is to analyze the specification established in the first iteration and define the ontologies for both the system component and the policy component of the models. In the system component, the ontologies are used to describe the information and data types that the systems store and process. In the policy component, the ontologies describe the entities, actors and objects that appear in the policies. The ontologies also store the patterns of the policies as well as the relations between the entities and the description of the entities. The ontologies established in this step serve as a meta model for the policy models and information flow models.

The main ontology elements for the system infrastructure include the following; the person ontology describing the patients and physicians, the document ontology that describes the document and information types of the system, including both the clinical and the administrative information. The administrative information models contain admissions, referrals, consents and treatment relations. The ontology element for the policies includes the information ontology describing the relevant information objects such as all sensitive information classes. It also includes the ontology of the actors of the policies and their relations.

The ontologies of the system component and the policy component are not completely independent of each other. The ontologies from both sides together control the behavior of the final system. The ontologies can be built such that they cover the concepts from both sides of the components or in later stages a mapping between the two can be created.

The ontologies are modeled in two steps. First the actor, object and other entity classes are defined together with their hierarchy. All other models, the pattern models and the policy models, use these classes directly or indirectly by referencing them. The pattern models are declared as a set of endpoints where each endpoint represents a role in the pattern. Each role has a class associated to it to define the set of objects that can be used in that endpoint.

The graphical model describing the pattern for an institutional policy is presented on Figure 4. This example institutional policy pattern enables the creation of policies that are a specialization of the sentence; *Requestor who is a Person can access an Object that is a Medical Document of a Patient who is a Person only if the specified Constraint is satisfied.* The institutional policy pattern also has two different semantic definition models anchored to it one for verification in formula and one for runtime enforcement in DROOLS [29]. Each of these semantic template models contains a set of logic program fragments in the respective language.

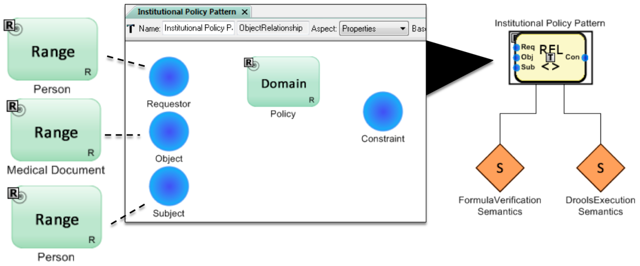

Figure 4: The Pattern model for institutional policies with two different semantics anchored.

4.4 Iteration 3: Information Flow and Policy modeling

The third iteration step is to create the information flow and policy models using the ontologies form iteration 2 based in the specification developed in iteration 1. The information flow model of the use case can be seen on Figure 5. The model closely follows the use case in how the sequence of visits gets modeled. This simple use case does not have request that would create a branching point so the request are all leafs in the sequence.

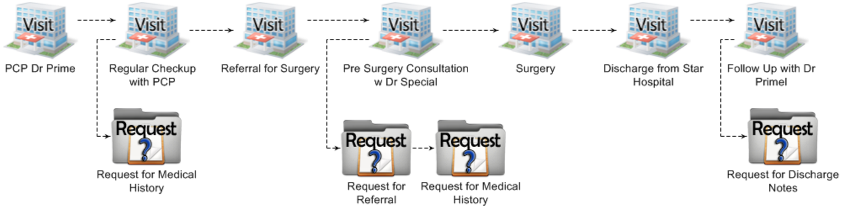

Figure 5: The information flow model of the use case

The system and information flow models enables the creation of use case models that represent a specific usage scenario of the modeled systems governed by the modeled polices. The information flow model is MDE extension of well known UML use-case model [9] and the Decentralized Information Flow Control Model [23].

The use case models are built using three distinctive sets of elements; the System and Actor models, as shown on Figure 3, a Sequence of episodes that describe a timeline of events, as shown on Figure 5, and the set of the governing policies. From the three, the references to the governing policies are the simplest. These models refer to one or more policy models that were specified using the patterns and ontologies. This explicit referencing makes it possible for the modeler to experiment with different policy sets and their effect on the use case.

The System and Actor models are exactly what their name implies. By explicitly specifying them gives a basic framework of the use case. There are two different actor types, Patient and Physician. Both actor models can be tagged with classes of the ontology giving the actor objects a rich description. The system models declare what systems the information flow is going through. Under each system model it is possible to specify what kind of information can be stored and the pattern of that information. The pattern of the information is specified the same manner as the policy patterns.

The actual information flows are specified by sequences of episodes. Each episode describes a set of transactions where new documents are being pushed into or retrieved from one of the systems. Request episodes can represent requests for a specific document, or documents of certain patient. The document stored in a system model is an instance of one of the patterns that are specified under the system model as a supported one. The document instance can use tags from the ontologies in the environment as well as any of the actor models defined in use case.

The policies gathered in Iteration 1 are also modeled in this step, these includes the government, the institutional and the data sharing policies. The more precisely the policies are modeled the closer the final system will be to the desired behavior. All the policies of the use case are modeled similarly to the external access policy of the data sharing agreement that is shown on Figure 6. The policy model consist of the actors of the policy the Patient and the External provider, which are individuals that are members of the Person class. The object of the policy is a Medical Document object that is from the Medical Document class or any of

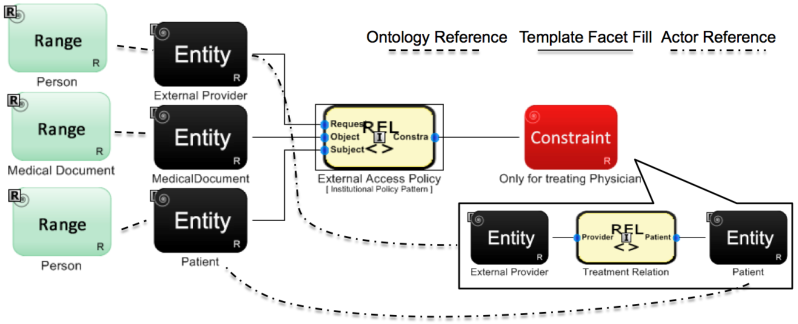

Figure 6: The Policy model for the policy specified in the data sharing agreement of the use case.

it descendants. The policy also has a constraint that constraints the disclosure of the Medical Documents to Requestors who are in a Treatment Relation with the Patient they requesting data on. All these models are connected to the Institutional Policy Pattern in the middle that provides the structural semantics to the other model components.

4.5 Iteration 4: Verification, composition and simulation

After the modeling of the information flows and the policies the next step is verify the policies, compose the policies with the information flow and system models and simulate the system behavior using the complete composition and check whether the models work according to expectations. In cases when the verification process discloses a problem, the developers have to track back to the previous iterations and fix the cause of the failure.

To verify the policy models, the policies have to be anchored to formal semantics. Such formal semantics can be provided using FORMULA as the language. With the formal semantics assigned to the models the policies can be checked for consistency, entailment and contradictions.

To compose the system models with the policy models the ontologies of them has to be made compatible. This is not necessary if the two models use the same ontology as described as option in Iteration 1. In the case when the ontologies are different it is necessary to map both the concepts and patterns from the policy models to the system models. Mapping the concepts can be done by introducing new “subclassOf” associations to the ontology between two concepts of the separate ontologies. The two concepts should be as closely related as possible. In case the two concepts are equivalent the association can be bi-directional. The mapping of the patterns are done similarly to the ontologies. In a separate mapping component of the information flow model the patterns that can be used in the information flow model in place of the patterns of the policy models are declared.

To simulate the complete models the pattern models have to be anchored to formal behavior semantics that is compatible of the simulation environment. The simulation environment has the behavior semantics of the information flow and system models statically implemented and the capability to interpret any of the syntactically correct models. The policy models are introduced into this simulation environment by composing them together with the pattern models and such providing formal semantics to them too. The simulation goes through all the episodes of the use case and checks all request models whether it is approved or denied when governed by the currently modeled set of policies. From the result of the simulation the correctness of the composed model can be deducted.

4.6 Iteration 5: System integration and executable policy code generation

Once the modeling is complete on both the system and policy component and the models have been verified, the last step is to put the models into use in the real system. The policy models can be used to generate executable code for policy enforcement or audit. The system and information flow models can be used to generate web services and their configuration to help system integration.

Using the semantic templates that were anchored to the pattern models it is possible to generate executable code from the policy models. Once the policy code is generated, the code has to be injected into a policy enforcement point. Using a policy enforcement point that supports policies in a logic programming language, like FORMULA or PROLOG, or a rule language like DROOLS' DRL [29] or ILOGs IRL makes this injection straightforward. It is also possible to generate high-level enforcement specification using XACML [22] specification. In cases when the enforcement point only supports "plain old" object oriented languages the injection of policy rules has to happen in a development cycle similar to a dependent library.

To help system integrators to incorporate the configured policy enforcement point the information flow models can serve as source go generate system or deployment configuration. These configuration files can help configure the web-services and other resources in the HIE system. The main services to be integrated are the policy enforcement point and the services that support its function. The most fitting target for code generation based on the system models is Web Service Business Process Execution Language (WS-BPEL) workflow specification language that can orchestrate all the system components presuming that all the necessary system components provide web-service interfaces.

5 Conclusion

In the chapters above we presented a novel MIC approach to model HIE systems together with the privacy polices that govern its functionality. We showed how taking iterative steps and refining the models of the system and the policies in each step would result in a design where there is no gap between the policies and the system infrastructure. The examples of through this publication also provide a good insight on how MIC tools can help different stakeholders to communicate and understand different aspects of health care systems. Our next step will be the full analysis and comparison of the presented approach to other well established methodologies. We also plan to create a set of tested and verified model libraries from widely adopted standards such as HL7 and the European EN 16306 to be used with modeling of the systems and data types.

Our greatest revelation has been the ambiguity and variation in institutional privacy policies. As state policies become more consistent and electronic health information exchange becomes a pre-requisite, institutions are each independently updating their policies to conform to their perception of law and to their own traditions. These deliberations consume many long hours but result in documents that are subject to variation in human interpretation and challenging to model explicitly in logic and ontologies. In addition, each EMR system still has different ways of representing identity, authorization, context, provenance, and other attributes critical to enforce privacy policies. Data standards will have to be enhanced and consistently enforced if policy enforcement in HIE is to extend beyond relatively tightly-coupled EMR systems.

6 Acknowledgments

The work presented in this paper was funded through Office of National Coordinator for Health Information Technology (ONC) Grant Number HHS 90TR0003/1. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the HHS.

References

[1] Rakesh Agrawal, Paul Bird, Tyrone Grandison, Jerry Kiernan, Scott Logan, and Walid Rjaibi. Extending relational database systems to automatically enforce privacy policies. In *Proceedings of the 21st International Conference on Data Engineering*, ICDE '05, pages 1013–1022, Washington, DC, USA, 2005. IEEE Computer Society.

[2] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Hippocratic databases. In *In 28th Intl Conference on Very Large Databases, Hong Kong*, 2002.

[3] A. Barth, J. Mitchell, A. Datta, and S. Sundaram. Privacy and utility in business processes. In *Computer Security Foundations Symposium, 2007. CSF '07. 20th IEEE*, pages 279 –294, july 2007.

[4] Adam Barth, Anupam Datta, John C. Mitchell, and Helen Nissenbaum. Privacy and contextual integrity: Framework and applications. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pages 184–198, Washington, DC, USA, 2006. IEEE Computer Society.

[5] Rafae Bhatti, Arjmand Samuel, Mohamed Eltabakh, Haseeb Amjad, and Arif Ghafoor. Engineering a policy-based system for federated healthcare databases. *IEEE Transactions on Knowledge and Data Engineering*, 19:1288–1304, 2007.

[6] Anupam Datta, Jeremiah Blocki, Nicolas Christin, Henry DeYoung, Deepak Garg, Limin Jia, Dilsun Kirli Kaynar, and Arunesh Sinha. Understanding and protecting privacy: Formal semantics and principled audit mechanisms. In *ICISS*, pages 1–27, 2011.

[7] PhD Ezekiel J. Emanuel, MD. Where are the health care cost savings? *JAMA, The Journal of the American Medical Association*, 307(1):39–40, 2012.

[8] Mark E Frisse, Kevin B Johnson, Hui Nian, Coda L Davison, Cynthia S Gadd, Kim M Unertl, Pat A Turri, and Qingxia Chen. The financial impact of health information exchange on emergency department care. *Journal of the American Medical Informatics Association*, 2011.

[9] Object M. Group. OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2. Technical report, Group, Object M., November 2007.

[10] Janine Hiller, Matthew S. McMullen, Wade M. Chumney, and David L. Baumer. Privacy and security in the implementation of health information technology (electronic health records): U.S. and EU compared. *Boston University Journal of Science Technology Law*, 17, 2011.

[11] Ethan K. Jackson, Tihamer Levendovszky, and Daniel Balasubramanian. Reasoning about metamodeling with formal specifications and automatic proofs. In Jon Whittle, Tony Clark, and Thomas Kuhne, editors, *MoDELS*, volume 6981 of *Lecture Notes in Computer Science*, pages 653–667. Springer, 2011.

[12] Ethan K. Jackson, Dirk Seifert, Markus Dahlweid, Thomas Santen, Nikolaj Bjørner, and Wolfram Schulte. Specifying and composing non-functional requirements in model-based development. In Alexandre Bergel and Johan Fabry, editors, *Software Composition*, volume 5634 of *Lecture Notes in Computer Science*, pages 72–89. Springer Berlin; Heidelberg, 2009.

[13] Ethan K. Jackson and Janos Sztipanovits. Formalizing the structural semantics of domain-specific modeling languages. *Software and Systems Modeling*, 8:451–478, 2009. 10.1007/s10270-008-0105-0.

[14] Gabor Karsai, Aditya Agrawal, and Akos Ledeczi. A metamodel-driven MDA process and its tools. In *WISME, UML 2003 Conference*, San Francisco, CA, October 2003.

[15] Peifung E. Lam, John C. Mitchell, Andre Scedrov, Sharada Sundaram, and Frank Wang. Declarative privacy policy: finite models and attribute-based encryption. In *Proceedings of the 2nd ACM*

SIGHIT International Health Informatics Symposium, IHI '12, pages 323–332, New York, NY, USA, 2012. ACM.

[16] Peifung E. Lam, John C. Mitchell, and Sharada Sundaram. A formalization of HIPAA for a medical messaging system. In *Proceedings of the 6th International Conference on Trust, Privacy and Security in Digital Business*, TrustBus '09, pages 73–85, Berlin, Heidelberg, 2009. Springer-Verlag.

[17] Akos Ledeczi, Miklos Maroti, Arpad Bakay, Gabor Karsai, Jason Garrett, Chuck Thomasson, Greg Nordstrom, Jonathan Sprinkle, and Peter Volgyesi. The generic modeling environment. In *Workshop on Intelligent Signal Processing*, Budapest, Hungary, May 2001.

[18] Janos Mathe, Jason B. Martin, Peter Miller, Akos Ledeczi, Liza Weavind, Andras Nadas, Anne Miller, David J. Maron, and Janos Sztipanovits. A model-integrated guideline-driven clinical decision support system. *IEEE Software, Special issue on Domain-Specific Languages & Modeling*, 2009. In Print.

[19] Elizabeth A. McGlynn, Steven M. Asch, John Adams, Joan Keesey, Jennifer Hicks, Alison DeCristofaro, and Eve A. Kerr. The quality of health care delivered to adults in the united states. *New England Journal of Medicine*, 348(26):2635–2645, 2003.

[20] D.L. McGuinness, F. Van Harmelen, et al. OWL web ontology language overview. *W3C recommendation*, 10:2004–03, 2004.

[21] Apurva Mohan and Douglas M. Blough. An attribute-based authorization policy framework with dynamic conflict resolution. In *Proceedings of the 9th Symposium on Identity and Trust on the Internet*, IDTRUST '10, pages 37–50, New York, NY, USA, 2010. ACM.

[22] Tim Moses. eXtensible Access Control Markup Language TC v2.0 (XACML), February 2005.

[23] Andrew C. Myers and Barbara Liskov. A decentralized model for information flow control. *SIGOPS Oper. Syst. Rev.*, 31(5):129–142, October 1997.

[24] Andrew C. Myers and Barbara Liskov. Protecting privacy using the decentralized label model. *ACM Trans. Softw. Eng. Methodol.*, 9(4):410–442, October 2000.

[25] Andras Nadas, Tihamer Levendovszky, Ethan K. Jackson, and Janos Sztipanovits. A model-integrated authoring environment for privacy policies. *Science of Computer Programming*, Special Issue on Success Stories in Model Driven Engineering, 2012. Submitted.

[26] H Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, (79):119–158, 2004.

[27] Hoangmai H. Pham, Deborah Schrag, Ann S. O'Malley, Beny Wu, and Peter B. Bach. Care patterns in medicare and their implications for pay for performance. *New England Journal of Medicine*, 356(11):1130–1139, 2007.

[28] Joseph Porter, Peter Volgyesi, Nicholas Kottenstette, Harmon Nine, Gabor Karsai, and Janos Sztipanovits. An experimental model-based rapid prototyping environment for high-confidence embedded software. In *20th IEEE/IFIP International Symposium on Rapid System Prototyping (RSP'09)*, Paris, France, 06/2009 2009.

[29] Mark Proctor, Michael Neale, Bob McWhirter, Kris Verlaenen, Edson Tirelli, Alexander Bagerman, Michael Frandsen, Fernando Meyer, Geoffrey De Smet, Toni Rikkola, Steven Williams, and Ben Truit. Drools, 2007.

[30] R. Bayardo T. Grandison C. Johnson R. Agrawal, D. Asonov and J. Kiernan. Managing disclosure of private health data with hippocratic databases. In *IBM Research White Paper*, 2005.

[31] Mark Strembeck. Testing policy-based systems with scenarios. In *Parallel and Distributed Computing and Networks*, volume 720, 2011.

[32] U.S.C. HIPAA: Health insurance portability and accountability act, 1996.

[33] U.S.C. Health information technology for economic and clinical health (HITECH) act, February 2009.