Published in the Policy Symposium,
Washington DC, July 2010

A Model for the Governance of Federated Healthcare Information Systems

Naftaly Minsky
Department of Computer Science, Rutgers University

Abstract

Modern healthcare is characterized by the increasing tendency for the health care records of a single patient to be dispersed throughout a complex network of health care providers. And some, or all, of such records, pertaining to a given patient, may have to be transfered to a provider to facilitate the treatment of this patient. Such transfer needs to be done quickly, because delays may adversely impacts the quality and cost of healthcare; and may, in some cases be a matter of life or death. But fast electronic transfer presents serious danger to the privacy and integrity of these records. This raises the need for governance, that is, for the formulation and enforcement of the societal policies and laws pertaining to the exchange of electronic healthcare records between the members of the often large and heterogeneous networks of healthcare providers.

This paper introduces a reference model for such governance, which has the following characteristics, among others: (a) decentralized, and thus scalable, enforcement mechanism; (b) seamless and secure interoperation between health care providers operating under different policies, and under different administrative domains; (c) support for the naturally hierarchical organization of the policies that govern the exchange of health care records; (d) the ability to change policies while the system governed by them continues to operate.

1 Introduction

Modern healthcare is characterized by the increasing tendency for the health care records of a single patient to be dispersed throughout a complex network of health care providers[1], including doctors' offices, clinics, hospitals, labs, pharmacies, medical research organizations, insurance companies, and government agencies. And some, or all, of such records, pertaining to a given patient, may have to be transfered to a provider to facilitate the treatment of this patient. Such transfer needs to be done quickly, because delays may adversely impacts the quality and cost of healthcare; and may, in some cases be a matter of life or death.

This need, and other factors, resulted in initiatives, all over the world, to create electronic healthcare records (EHRs), and to facilitate their rapid transfer among providers. But such rapid transfer raises serious issues of privacy and integrity of these records. In the US, in particular, the HIPAA federal legislation mandates standards for privacy and security of EHRs, resulting in policies and laws about the exchange of EHR's, which must be met throughout the US—and analogous legislation exists in many other countries. Moreover, besides such a global HIPAA-based policy, a given state may have its own policy regarding the sharing of EHRs generated by its providers. Furthermore, individual healthcare providers in a given geographic region tend to form coalitions whose purpose is to facilitate exchange of EHRs among their members. And such a coalition—often called HIE, for "healthcare information exchange"—is likely to have its own policy regarding the sharing of its members' EHRs with others.

The country wide network of healthcare providers has come to be known in the US as Nationwide Health Information Network (NHIN) [19]. This large and heterogeneous network, and its governance—namely, the formulation of its various policies, their maintenance, and their enforcement—has the following challenging characteristics[2]. (We label these characteristics "C1" to "C5", for reference purpose.)

C1 *Conformance hierarchy of policies:* As we have seen above, the process of EHR exchange is subject to a multiplicity of policies, which form what we call, *conformance hierarchy*. In particular, the state policies concerning the exchange of EHRs must conform to the global HIPAA policy; and the policy of a coalition must conform to the policy of the state in which it resides—and thus transitively, it must conform to the HIPAA policy. (We will say more about this hierarchy in due course.)

[1] We use here the term healthcare provider, or simply "provider," for a system that generates and/or maintains healthcare records, whether it actually provides healthcare—like a doctor's office, or a hospital—or has a secondary healthcare role like that of an insurance company.

[2] Shared by analogous healthcare networks in other countries.

C2 *Interoperation:* Arbitrary pairs of providers, which may belong to different coalitions at different states—and thus operate under different policies—may need to *interoperate* in order to exchange EHRs.

C3 *The sheer complexity of policies:* In particular, policies that govern the exchange of EHRs need to be sensitive to the content of the EHR being exchanged, and to the history of exchange of EHRs (i.e., they need to be stateful); and they need sometimes be proactive. Stateful policies are required, for example, to regulate the dynamic conversations between the various providers; and to detect and report violations of quality of service (QoS) requirements. And proactive policies are required to mandate such things as timely reporting and logging, among other things.

C4 *The scale of the EHR-exchange process:* The number of providers exchanging EHR's, and the number of EHRs being exchanges, are very large.

C5 *The evolution of policies, while the system operates:* The various policies in the conformance hierarchy that governs the process of EHR-exchange are bound to change over time, mostly independently of each other. The resulting process of evolution of the conformance hierarchy must be carried out while the exchange of EHR's goes on—because such an exchange cannot be stopped over the entire system of healthcare providers. We refer to this kind of evolution as *in vivo* (i.e., evolution in a living organism, as it were).

These characteristics of NHIN—and others, to be mentioned later—present a formidable challenge to its governance. And, to the best of my knowledge, this challenge has not been met by the various recent attempts—such as [5, 12, 23, 19]—to implement a governance mechanism for NHIN. Meeting this challenge is the objective of this paper.

In designing a governance mechanism for NHIN it is important to realize that the heterogeneity of the healthcare provider, and of the software systems that manages their EHRs, implies that one cannot trust all these systems to enforce the policies they are subject to. Rather, the enforcement needs to be carried out by some kind of secure *policy based middleware* (or PBM, for short), which is to be used to mediate the interaction between the various healthcare providers, imposing the appropriate policies on the exchange of EHRs between them. The choice of such a PBM, which can cope with the characteristics of NHIN listed above, is therefore critical to its governance.

Our choice of a PBM, and the motivation for it, are described in Section 2. Using this PBM we introduce in Section 3 a reference model for the governance mechanism of NHIN. We then discuss in Section 4, the the implementation status of this model, and outline the open problems that need to be addressed for this model to be applied to a real NHIN. And we conclude in Section 5 by pointing out the broader implication of this work.

2 The Choice of Policy Based Middleware

We start this section by spelling out the properties that a PBM needs to satisfy in order to serve as the basis for the governance of an NHIN. Then, in Section 2.1 we describe briefly the LGI middleware, which satisfies these properties, and is therefore our choice for the governance of NHIN. And we conclude, in Section 2.2, with a brief review of related PBMs.

To cope with the challenging characteristics of NHIN listed in the Introduction, we maintain that a policy based middleware used as the basis for the governance of NHIN should satisfy the following set of properties.

1. *Support for enforced conformance hierarchy of policies:* The PBM needs to be able to organize policies into a conformance hierarchy. Furthermore, the transitive conformance relation underlying a hierarchy needs to be enforced. By this we mean that if a policy P is defined as subordinate to some policy Q in the hierarchy, then P should be guaranteed to conform to Q—and this guarantee should be automatic, rather than based on human testing and assertions. Such enforcement is needed, in particular, so that no coalition of healthcare providers would be able to insert its *coalition-policy* into the policy hierarchy, unless it does, in fact, conform to the global HIPAA policy. More about the need for the policy hierarchy to be enforced, in due course.

2. *Seamless interoperability between policies belonging to the same hierarchy:* Due to characteristic C2 of NHIN, the PBM must provide for secure and seamless means for providers, operating under different policies in a hierarchical ensemble, to interoperate. By "seamless" we mean, informally speaking, that a pair of policies belonging to the same hierarchy does not need to be *sewn*, or composed, together in any way, for them to be able to interoperate. If this is possible then a pair of providers belonging to different coalitions may be able to interoperate easily on the basis that both of them observe the given HIPAA-based policy.

3. *High expressive power:* Given characteristic C3 of NHIN, the PBM should enable the formulation and efficient enforcement of a wide range of policies, including stateful and proactive policies, and policies that are sensitive to the content of messages.

   Decentralization of the enforcement of policies: Given characteristic C4 of NHIN, decentralization of en-

forcement is critical for ensuring that the system can scale with the amount of message traffic. (Note that as argued in [16], replication of a reference monitor is not a practical decentralization technique whenever policies tend to be stateful.) Moreover, as we shall see in Section 3.2, decentralization is also facilitates seamless interoperability.

4. *In vivo evolution:* The PBM should support the in vivo evolution of hierarchical law ensembles, which is problematic in this case due to the required decentralized enforcement (c.f. [21].)

In order to satisfy these properties, we have chosen the LGI policy based middleware to be used in our model of NHIN. Although LGI, and various applications of it, have been published in various journals and conferences, we include here a brief overview of it for completeness.

2.1 The Law-Governed Interaction (LGI) Middleware—an Overview

Broadly speaking, LGI is a governance middleware that enables an open and heterogeneous group of distributed *actors* to engage in a mode of interaction *governed* by an explicitly specified and strictly enforced policy, called the *law* of this group. By "actor" we mean an arbitrary process, whose structure and behavior is left unspecified; but an actor engaged in an LGI-regulated interaction, under a law $\mathcal{L}$, is called an $\mathcal{L}$-$agent$, and the messages sent by an $\mathcal{L}$-agent are called $\mathcal{L}$-messages.

LGI thus turns a set of disparate actors, which may not know or trust each other, into a *community* of agents that can rely on each other to comply with the given law $\mathcal{L}$—this is called an $\mathcal{L}$-*community*. This is done via a distributed collection of trustworthy generic components called *private controllers*, one per $\mathcal{L}$-agent, that mediate all interaction between these agents, subject to the specified law $\mathcal{L}$ (as illustrated in Figure 1). All told, LGI goes well beyond conventional access control, in its ability to cope with the increasing size, openness, and heterogeneity of open systems. It is, in particular, inherently decentralized, and thus scalable even for a wide range of stateful policies. And it is very general. A prototype of LGI has been recently released and is being used in academia, and even in some industry. This section provides only a very brief overview of LGI, hopefully sufficient for understanding the gist of this proposal. For more information, the reader is referred to the LGI tutorial and manual [15], and to a host of published papers.

Agents and their Private Controllers: An $\mathcal{L}$-agent x is a pair $x = \langle A_x, T_x^{\mathcal{L}} \rangle$, where A_x is an *actor*, and $T_x^{\mathcal{L}}$ is its *private controller*, which mediates the interactions of A_x with other LGI-agents, subject to law $\mathcal{L}$. Each controller $T_x^{\mathcal{L}}$ maintains the *control state* (or, "cState") of agent x, which is some function of the history of interaction of x with other community members. The nature of this function, and its effect on the ability of x to communicate, is largely defined by the law $\mathcal{L}$. The concept of law is defined in the following section. The role of the controllers is illustrated in Figure 1, which shows the passage of a message from an actor A_x to A_y, as it is mediated by a pair of controllers, first by controller $T_x^{\mathcal{L}}$, operating under law $\mathcal{L}$, and then by $T_y^{\mathcal{L}}$, operating under law $\mathcal{L}'$, which may or may not be identical to $\mathcal{L}$—note that LGI provides for interoperability between different laws. (We will have more to say later about the pair of controllers mediating this interaction.)

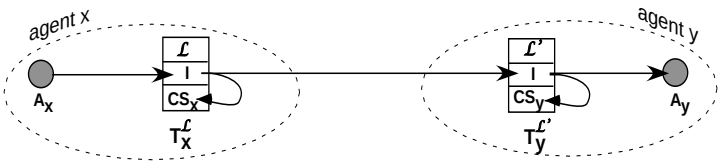

Figure 1. *Interaction between a pair of LGI-agents, mediated by a pair of controllers*

The Concept of Law Under LGI: An *LGI law* (or, simply, a *law*) is defined in terms of three elements: (a) a set E of *regulated events*; (b) a set O of *control operations*; and (c) the *control-state* (CS_x) associated with each agent x. More specifically, E is the set of events—such as the sending and arrival of a message—that may occur at any agent, and whose disposition is subject to the law. O is the set of operations that can be mandated by a law, to be carried out at a given agent, upon the occurrence of regulated events at it. In a sense, these operations constitute the *repertoire* of the law—i.e., it is the set of operations that the law is able to mandate. This set includes operations like forwarding a message, and updating the state of a given agent. Finally, the *control-state*, or simply the state, of an LGI agent is the state maintained by the controller of this agent agent, which is distinct from the internal state of the actor of that agent. This state, which is initially empty, can change dynamically in response to the various events that occur at it, subject to the law under which this agent operates.

Now, The role of a law under LGI is to decide what should be done in response to the occurrence of a regulated event at an agent operating under this law. This decision, which is called the *ruling of the law*, consists of a sequence of zero or more control operations from the set O. More formally, a law is defined as follows.

Definition 1 *Given a set E of all regulated events, a set O of all control operations, and a set S of all possible control-*

states, a law $\mathcal{L}$ is a function: $\mathcal{L} : E \times S \rightarrow O^*$

In other words, a law maps every possible $(event, state)$ pair into a sequence of zero or more control operations, which constitute the *ruling* of the law.

Note that this definition does not specify a language for writing laws. This for several reasons: First, because despite the pragmatic importance of choosing an appropriate law-language, this choice has no impact on the semantics of the model itself, as long as the chosen language is sufficiently powerful to specify all possible functions of the form of Definition 1. Second, by not specifying a law-language we provide the freedom to employ different law-languages for different applications domains, possibly under the same mechanism. Indeed, the implemented Moses mechanism employs two different law-languages, one based on the logic-programming language Prolog, and the other based on Java. Both of these languages are Turing complete, and can therefore specify arbitrary LGI laws. One can also use more restricted languages, if one is interested only in a subset of laws characterized by Definition 1 above.

Our Prolog law language, which is essentially and ECA language, is exemplified by a very simple law introduced below.

The Local Nature of LGI Laws, and their Global Sway: One important characteristic of LGI laws is that they are inherently local. Without going into technical details, locality means that an LGI law can be complied with, by each member of the community subject to it, without having any direct information of the coincidental state of other members. This locality is a critical aspect of LGI for two major reasons: First, because locality is necessary for decentralization of law enforcement, and thus for scalability even for stateful policies. And second, because locality facilitates interoperability between different laws, and enables the construction of law-hierarchies, as has been shown in [3].

Remarkably, although locality constitutes a strict constraint on the structure of LGI laws, it does not reduce their expressive power, as has been proved in [15]. In particular, despite its *structural locality*, an LGI law can have *global effect* over the entire $\mathcal{L}$-community—mostly because all members of that community are subject to the same law—and can, thus, be used to establish *mandatory*, community wide, constraints.

The Dual Mediation of Communication Under LGI: One of the significant aspects of LGI is that it involves duel mediation of every exchange of messages between agents: one on the side of the sender of a message, and one on the side of its receiver. Specifically, the passage of a message from an actor c_x of an $\mathcal{L}$-agent x to an actor c_y of an $\mathcal{L}'$-agent y, must be mediated first by the controller $T_x^{\mathcal{L}}$ associated with c_x, and then by the controller $T_y^{\mathcal{L}'}$, associated with c_y, as is illustrated in Figure 1. This is a direct consequence of the locality of LGI laws, which requires both the sender and receiver to individually comply with the law under which each of them operates. This duel mediation is in contrast with the conventional access control mechanisms, which use a single *reference monitor* to mediate the exchange of messages. Such a reference monitor is usually placed at the server side, or is used as a central *policy decision point* (PDP) for an entire system, as under XACML [10].

The dual mediation under IC has several important implications, not the least of them is that it facilitates interoperability by providing flexible control over cross-interaction between agents operating under different laws. Moreover, as has been shown in [16], the duel control turns out to be more efficient than centralized control, in many circumstances.

On the Basis for Trust Provided by LGI: The interaction (i.e., message exchange) between actors operating under a given law $\mathcal{L}$ can be trusted to conform to $\mathcal{L}$, even if the actors themselves have no trust in each other. What needs to be trusted here are the controllers that mediates this interaction. In other words the set of controllers being used by a given system of actors serve as a trusted computing base (TCB) of that system (or, more precisely, this set of controllers constitutes a *distributed trusted computer base*, or DTCB.

This DTCB is provided under LGI by means of what is called a *controller service* (CoS). This is an organization that implements and maintains a distributed collection of generic controllers, each of which can operate for any actor, and under any given law. Arguably, such a generic DTCB can, in principle, be more dependable and more trustworthy—or, if you will, more fault tolerant and more secure—that traditional, centralized, TCB. A prototype of a CoS has been implemented.

The Organization of Laws into *Conformance Hierarchies*: LGI enables its laws to be organized into what we call *conformance hierarchies*. Each such hierarchy, or tree, of laws $t(\mathcal{L}_0)$, is rooted in some law $\mathcal{L}_0$. Each law in $t(\mathcal{L}_0)$ is said to be (transitively) *subordinate* to its parent, and (transitively) *superior* to its descendants. And, given a pair of laws $\mathcal{N}$ and $\mathcal{M}$ in $t(\mathcal{L}_0)$, we write $\mathcal{N} \prec \mathcal{M}$ if $\mathcal{N}$ is subordinate to $\mathcal{M}$. Semantically, the most important aspect of this hierarchy is that if $\mathcal{N} \prec \mathcal{M}$ then $\mathcal{N}$ *conforms* to $\mathcal{M}$, in the sense that *law $\mathcal{N}$ satisfies all the stipulations of its superior law $\mathcal{M}$*.

This is a much more general concept of conformance than adopted by some policy mechanisms (see [7], for example), where a policy P is considered in conformance

with a policy Q, *only* if P is at least as restrictive as Q. Briefly, the LGI's concept of *conformance hierarchy* has two key properties: (a) it is *heterogeneous*, and (b) it is *enforced*. The hierarchy is heterogeneous with respect to conformance, in the following sense: every law in the hierarchy can specify the degrees of freedom it leaves to its subordinate (descendant); that is, each law circumscribes the degree and manner in which its descendant can deviate from it. And the hierarchy is enforced by its very construction. That is, the very definition of a law $\mathcal{N}$ as subordinate to $\mathcal{M}$, prevents $\mathcal{N}$ from violating the restriction imposed by $\mathcal{M}$ on its subordinates. The manner this is done has been defined in [3], and it is too complex to describe here.

Other Features of LGI, and its Performance: We will list here some of the notable features of LGI, which we were not able to discuss in this short overview, and will provide references to them for the interested reader. These features are: (1) the concept of *enforced obligation*, that provides LGI with important proactive capabilities; (2) the treatment of *exceptions*, which provides LGI with fault tolerance capabilities; (3) the treatment of *certificate*, which is obviously necessary for the regulation of distributed computing; and (4) *interoperability* between different laws, even if they do not belong to the same hierarchy. All this, and the performance of LGI, is discussed in the LGI Manual [15].

2.2 Related Policy Based Middlewares

Although to my knowledge there is no PBM, besides LGI, that supports all the features listed at the top of this section as necessary for NHIN, certain PBMs do support some of this features, at least to some extent. The following is a brief—due to lack of space—acknowledgment of some of these mechanisms.

First, regarding conformance hierarchy of policies: Oasis [4] introduced the concept of *meta policy*, which is similar to our conformance hierarchy in that it is to be conformed to by all sub-policies that regulates different parts of a given system. However, conformance to the meta policy is not enforced, and needs to be verified manually. (Ponder [7] also features a concept of meta-policy, but it mostly deals with conflict and relationships between policies.)

Second, interoperability between policies has been addressed in several papers, such as [14], essentially by composing a pair of interoperating policies into a single policy that is consistent with both. This, however, is far for seamless, and is particularly problematic for large scale federations. (More about this in Section 3.2.)

Third, regarding stateful policies; i.e., policies that are sensitive to the history of interaction. Although specific types of such policies, such as *Chinese Wall* and *dynamic separation of duties* policies , have been known for a long time, only recently we have seen mechanisms that support general treatment support of stateful policies. Perhaps the best of these are the proposal due to Jojodia et al., in their 2001 paper [11], and the SPL mechanism introduced by Ribeiro et al. [18]—by neither of these is as scalable as LGI, due to their centralized enforcement mechanism.

Finally, regarding locality of control, one should mention trust management [6]. But trust management has a fundamental limitation with respect to the application at hand: it is basically *server centric*, in the sense that every server is free to write its own policy about which kind of credentials it is willing to accept, and how to interpret them. In particular, it provides no means for ensuring that two health care providers observe the same global policy.

3 A Reference Model for National Health Information Networks (NHINs)

Broadly speaking, an NHIN is modeled here as a four-tuple $\langle H, P, LE, T\rangle$, whose components are defined below.

H is a set of healthcare providers, in our broad sense of this term. These providers are assumed to be grouped into a disjoint set of regional coalitions. It should be pointed out that a coalition can have some servers used to monitor and manage the activities of its members, we will not distinguish here between these servers and real healthcare providers.

P is the set of patients whose EHRs are maintained by any of the providers in H; LE is the hierarchical ensemble of laws that collectively govern the flow of messages between the various parts of this systems. More about which below. Finally, T is a set of generic LGI-controllers, maintained by the distributed *controller service* (CoS) that serves as the trusted computing base for the NHIN. That is, the interaction between providers and patients in an NHIN will be mediated by these controllers, subject to various laws in LE, as we shall see.

The rest of this section is organized as follows: Section 3.1 describes the law-ensemble LE, which is, in many ways, the heart of our model; Section 3.2 discusses the *seamless interoperability* between providers resulting from this model; Section 3.3 discusses the treatment under this model of patients policies regarding the exchange of their own EHRs between providers; and Section 3.4 reviews some related models of NHIN.

3.1 The Hierarchical Law Ensemble (LE) of an NHIN

The following is a schematic and simplified description of the law ensemble that governs an NHIN, as depicted in Figure 2. This ensemble of laws is organized into an *enforced conformance hierarchy*, a concept introduced in the

context of LGI, and discussed in Section 2.1. (This hierarchy is simplified, in particular, by ignoring the state policies, which are likely to be included in a real hierarchy, at its second level.)

The root of this hierarchy is law $\mathcal{L}_G$, which, due to the conformance relation underlying this hierarchy, has global dominion over the entire NHIN. That us, this law governs the exchange of EHRs between all providers in H, representing such things as the regulations promulgated in the US in response to the HIPAA act. At the second level of this hierarchy we have the following set of laws. First, there is a set $\{\mathcal{L}_i\}$ of the laws of the various coalitions of providers. Second, there is law $\mathcal{L}_P$, under which the patients are to operate. And third, there can be any number of laws $\{\mathcal{L}_{P_j}\}$ that are subordinate to $\mathcal{L}_P$, which represent different policies that patient may adopt for the exchange of their EHRs. We now illustrate the nature of this hierarchy with some examples of the provisions one may find in laws at different levels of it. These example are stated informally, and abstractly.

The Global Law of an NHIN: One can expect the following kinds of provisions to be made by the root law $\mathcal{L}_G$.
(1) *Establishing global foundation for trust:* In particular, one may impose the following provision. All actors operating within the coalition must authenticate themselves via certificates signed by CAs, which are themselves authenticated by a global CA specified by this policy. So, in particular, the CA of a given provider must be authenticated by this global CA.
(2) *Establishing the structure and semantics of the message exchange between providers:* In particular this law may require all such messages to identify their sender via its name, address, etc., as asserted by the above mentioned certificates.
(3) *If asked by a patient for his/her record,* a provider must send these records within a reasonable (unspecified here) time period. (This is an example of quality of service (QoS) provisions.)
(4) *No healthcare records about a patient should be sent without that patient's consent*. (We will discuss later the manner in which such consent can to be obtained.)
(5) *Establishing and securing audit trails:* This law may require that certain transactions should be logged in specified logs, establishing access control to these logs. (The importance of auditing for the security of health care records has been discussed in [8], along with a logic for the analysis of the information collected in audit trails.) Recall that a simple example of establishing an audit trail is given in Figure **??**
(6) *Providing subordinate policies (e.g., coalition policies) with the ability to override certain provisions of this law, if such overrides are done according to a specified protocol*.

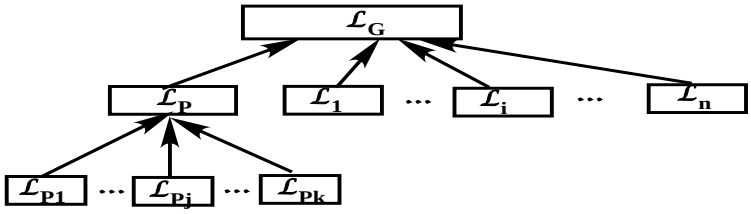

Figure 2. A Sketch of the Hierarchical Law-Ensemble of an NHIN

For example, an override of point 4 above may be allowed under emergency condition, if authorized by a proper local authority, and if properly documented.

Laws of Coalitions of Providers: One can expect the following types of provisions to exist in the law $\mathcal{L}_i$ of a typical coalition C_i.
(1) Establishing a local audit trail, beyond of what is required by law $\mathcal{L}_G$.
(2) Specification of who is allowed to carry out which kinds of overrides of the provision of a superior law (where the superior law allows such overrides).
(3) Establishing quality of service (QoS) requirements for EHR-exchange between member of the coalition at hand, which are at least as stringent as those required by $\mathcal{L}_G$.

The Patient Law $\mathcal{L}_P$: This law may play two different roles. First, it can be the law under which patients would operate in order to request their own records from various providers. In particular, this law could specify the type of credentials that every patient is required to present so that their request would be honored. However, it may, actually, be simpler for the global law, $\mathcal{L}_G$, to specify the credentials that patients need to provide, and let patients communicate with providers directly via HTTP or SOAP protocols (this possibility is provided by LGI).) The second, and very important function of law $\mathcal{L}_P$, and of its subordinate laws, will be discussed in Section 3.3.

On the Specification of Law Hierarchies: As has been pointed out, LGI provides for the specification of conformance hierarchies of laws, via its law language, and for the enforcement of the conformance relation underlying such hierarchies. But space limitation preclude the formalization of the example hierarchy described above, by defining it via the LGI law language. The reader is referred, instead, to [3], which introduces the concept of conformance hierarchy in detail, illustrating it with an example of some complexity. Moreover, the example given in that paper, has some similar

elements—such as auditing and the use of certificates—to the example hierarchy introduced above.

3.2 Seamless Interoperation

A remarkable, and very important, property of such hierarchical organization of laws is that it provides for a seamless interoperation between providers operating under different laws in the hierarchy. In particular, a pair of providers x and y, that belong to two different coalitions C_i and C_j, respectively, and thus governed by different laws $\mathcal{L}_i$ and $\mathcal{L}_j$, can interoperate with each other with confidence that they both conform to the global law $\mathcal{L}_G$. This is because under LGI each of them can detects cryptographically[3] that the other's law is subordinate to the same global law $\mathcal{L}_G$, and therefor conforms to it, because the conformance is enforced by the hierarchy. And it is worth emphasizing that this confidence in the mutual conformance to law $\mathcal{L}_G$ is independent of the specific laws under which they operate, except they they both are subordinate to $\mathcal{L}_G$.

To appreciate the importance of such seamless interoperability, consider the conventional *composition-based* approach [14] to interoperability between two different policies[4] P_1 and P_2. Under this approach it is necessary to *compose* policies P_1 and P_2 into a single policy P_{12} that is consistent with both of these policies, and then to have P_{12} enforced by a reference monitor mediating the interoperation in question. Note that for providers belonging to the various coalitions to be able to interoperate with each other one needs a quadratic number (in terms of the number of coalitions) of such compositions—a truly daunting prospect. In addition, composition of policies tends to be computationally hard [14], even for relatively simple policy languages; and it is often impossible to carry out a composition due to an inherent conflict between the policies in question. Clearly, interoperation based on composition is anything but seamless.

Moreover, the need to form a pairwise composition of policies for every pair of coalitions whose members may need to interoperate would introduce a serious rigidity to the entire system, making it very hard for individual coalitions to create their policies, or to change them.

It is worth pointing out that there are two technical reasons for not needing composition to interoperate under LGI. The first is the enforced nature of our conformance hierarchy; and the second reason is the decentralized nature of communication under LGI, with its dual control over every message exchange—which allows every provider to operate under its own law, knowing nothing about the law of its interlocutor, except that it belongs to the same conformance hierarchy (c.f. [3]).

3.3 Supporting the Cross-Cutting Authority of Patients

One of the basic provisions of HIPAA, and of analogous laws in other countries, is that any transfer of an EHR of a given patient p generally requires the consent of p. This overwhelming power invested in the patients with respect to their EHRs has been characterized in [9] as "Perhaps the most challenging aspect" of EHR exchange. In the context of this model, the problem is that the patient's policy has a kind of cross cutting authority with respect to the neat hierarchical law structure we presented so far. In other words, every exchange of an EHR is now subject not only to the laws of the two parties to such an exchange, but also to the patient's policy.

We deal with this problem in the following manner. As depicted in Figure 2, law $\mathcal{L}_P$ can have any number of subordinate laws that are meant to represent patient's policies regarding the exchange of their own EHRs between pairs of providers. Some of these laws may be formulated by experts, to fit common patterns of patient's consent to EHR transfer; others may be formulated explicitly by patients. In any case, patients will notify the holders of their EHRs of their choice of one of these patient's laws. And the global law $\mathcal{L}_G$ will force each provider to consult this patient law before sending any of the patient's EHRs. (It should also be pointed out that one can expect a deeper hierarchy emanating from law $\mathcal{L}_P$, than depicted in Figure 2.)

3.4 Related Work

One of the earliest attempts at designing a governance mechanism for the exchange of EHR, by Anderson [1], recognized the need for local control. But there are two important differences between that approach to local control and ours. First, Anderson describe local control as follows: systems that handle personal health information shall have a subsystem that enforces the above principles in an effective way. Its effectiveness shall be subject to evaluation by independent experts But placing the would be TCB inside the system to be protected by it, is not very trustworthy. And is arguably less secure than ensuring that the LGIs set of generic controllers, which are not included in any of the systems of providers, are are operating correctly. Second, this technique cannot support any equivalent of the conformance hierarchy of LGI, which, as has been argued in this paper, is essential for the governance of EHR exchange.

[3] This is done roughly as follows: each of the controllers serving x and y is built to maintain the hashes of the sequences of laws that constitute the entire pedigree of the law this controller operates under. These sequences of hashes are exchanged by the controllers serving x and y during their handshake, so they can determine whether or not they operate under a common superior law—say, the HIPAA law—and thus conform to it.

[4] We are using the term "policy" here, again, because this is the term used in this research.

By and large, even the more recent attempts at the governance of EHR exchange within an NHIN-like network, such as [5, 12, 23, 19], seem unconcerned about the characteristics of an NHIN identified in Section 1. In particular, none of them supports a conformance hierarchy of policies. Even the semi official draft specification of the NHIN authorization framework issued by the US department of Health and Human Services [19], deals only with the exchange of EHRs between *communities* (which we call here "coalitions") subject to HIPAA policy, providing no means for ensuring that the exchange within the various community conform to HIPAA—this despite the fact that the vast majority of EHR exchange is within communities, not between them.

Moreover, most conventional approaches to the governance of EHRs employ mechanisms such as RBAC [20], X-GTRBAC and XACML, despite the lack of support for stateful and proactive policies by these mechanisms. Yet, some researchers complain about certain limitations of conventional AC mechanisms. For example, [12] complains about the inflexibility of RBAC, and its lack of state; and [9] points out the need to ensure proper reporting, which requires proactive capabilities such as the concept of obligations in LGI. Finally, in areas closely related to the governance of NHIN, Joshi et al. [13] state that "secure interoperability between institutions poses a major challenge which has not been met so far," referring to just one of the qualities we are concerned with.

4 State of Implementation, and some Open Problems

A prototype of our reference model for NHIN has been implemented, as a proof of concept. Using the current version of LGI, we have implemented most of the types of provisions outlined in Section 3 on a very small scale, and without any real healthcare record. The implementation of a more substantial prototype, with real, but anonymized, EHRs, is yet to be carried out. In addition, the potential use of this model to support NHINs raises several open, or partially open, technical problems that need to be addressed. The following is a brief outline of two of these problems.

In Vivo Evolution of the Hierarchical law ensemble: The reason that in-vivo evolution of laws (i.e., evolution of laws while the system governed by them continues to operate) is problematic in our case, is the decentralized nature of our enforcement mechanism. This means, in particular, that the controllers mediating the operations of a set of agents under a given law $\mathcal{L}$ need to be updated in a *logically atomic* manner. We recently solved this problem [21] for the case of system governed by a single law. We now need to extend this solution for changes made in a law-hierarchy.

Dealing with uncertain lifetime of digital certificates: Due to the heterogeneity and size of HIEs, their trust structure needs to be based on digital certification. But unlike diamonds, certificates are not forever. They may have a specified expiration date, and might be revoked, for various reasons, at unpredictable times. The question is what effect the expiration or revocation of a certificate should have on the privileges of the participant previously authenticated by it. This is quite a critical problem, which is rarely discussed in the literature. We have addressed this problem in [2], and solved it via LGI in the context of a single healthcare provider, operating under a single law. The challenge would be to extend this solution to our much more complex context of NHIN.

5 Conclusion

The model proposed here for NHIN has a much broader significance. This is because the challenging characteristics of NHIN, for whose support our model has been introduced, are shared by other type of distributed systems. This is true, in particular, for large enterprise systems, for virtual organizations (VOs), for grid computing, and for many SOA-based systems. All these types of systems tend to be very large and heterogeneous; they need to be governed by by a multitude of policies—regulating various parts of the system, and various aspects of it; and such policies are naturally hierarchically organized [22]. Moreover, like in the case of NHIN, the unavoidable evolution of the policies of such systems must often be carried out in vivo.

One can expect, therefore, that models broadly resembling the one introduced in this paper would be suitable for the above mentioned type of systems, and that these models would have to be based on a PBM like LGI.

As a partial support for this prediction, I conclude with the following citation from a paper about grid computing [17]—whose authors include Foster and Kesselman, the founders and leaders of this area of research—*"For highly-sensitive applications where greater assurance of resource enforcement of community policy is required, a mechanism such as Law-Governed Interaction can be used."*, referring to the LGI middleware.

Acknowledgment: I am grateful to David Vickers, from the Southwest Research Institution, for many useful discussion about the EHR problem.

References

[1] J. R. Anderson. A security policy model for clinical information systems. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 1996.

[2] X. Ao, N. Minsky, and V. Ungureanu. Formal treatment of certificate revocation under communal access control. In *Proc. of the 2001 IEEE Symposium on Security and Privacy, Oakland California*, May 2001.

[3] X. Ao and N. H. Minsky. Flexible regulation of distributed coalitions. In *LNCS 2808: Proc. European Symp. on Research in Computer Security (ESORICS)*, Oct. 2003.

[4] A. Belokosztolszki and K. Moody. Meta-policies for distributed role-based access control systems. In *Proc. of the IEEE 3rd International Workshop on Policies, Monterey, California*, pages 106–15, June 2002.

[5] R. Bhatti, K. Moidu, and A. Ghafoor. Policy-based security management for federated healthcare databases (or rhios). In *Proc. of the international workshop on Healthcare information and knowledge management*, pages 41–48, June 2006.

[6] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The role of trust management in distributed systems security. *Secure Internet Programming: Issues in Distributed and Mobile Object Systems*, 1603, 1999.

[7] N. Damianou, N. Dulay, E. Lupu, and M. Sloman. The ponder policy specification language. In *Proc. of Policy Worshop, Bristol UK*, January 2001.

[8] M. Dekker and S. Etalle. Audit-based access control for electronic health records. *Electronic Notes in Theoretical Computer Science*, (168):221–236, 2007.

[9] D. M. Eyers, J. Bacon, and K. Moody. Oasis role-based access control for electronic health records. In *IEEE Proceedings-Software*, pages 16–23, 2006.

[10] S. Godic and T. Moses. Oasis extensible access control. markup language (xacml), version 2. Technical report, Oasis, March 2005.

[11] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subramanian. Flexible support for multiple access control policies. *ACM Trans. on Database Systems*, 26(2), June 2001.

[12] J. Jin, G. J. Ahn, H. Hu, M. J. Covington, and X. Zhang. Patient-centric authorization framework for sharing electronic health records. In *Proc of the ACM Symp on Access Control Models*, June 2009.

[13] J. Joshi, A. Ghafoor, V. Aref, and E. Spafford. Digital government security infrastructure design challenges. *IEEE Computer*, pages 66–72, January 2001.

[14] P. McDaniel and A. Prakash. Methods and limitations of security policy reconciliation. In *Proc. of the IEEE Symp on Security and Privacy*, May 2002.

[15] N. H. Minsky. *Law Governed Interaction (LGI): A Distributed Coordination and Control Mechanism (An Introduction, and a Reference Manual)*, February 2006. (available at `http://www.moses.rutgers.edu/`).

[16] N. H. Minsky and V. Ungureanu. Law-governed interaction: a coordination and control mechanism for heterogeneous distributed systems. *TOSEM, ACM Transactions on Software Engineering and Methodology*, 9(3):273–305, July 2000.

[17] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. A community authorization service for group collaboration. In *Proc. of the IEEE 3rd Int Workshop on Policies for Distributed Systems, Monterey, California*, June 2002.

[18] C. Ribeiro and P. Ferreira. A policy-oriented language for expressing security specifications. *International Journal of Network Security*, 5(3):299–316, November 2007.

[19] F. Richard. Nationwide health information network (nhin) authorization framework v2.2. Technical report, Department of Health and Human Services, US Gov., September 2009. website: `http://healthit.hhs.gov/portal`.

[20] R. Sandhu, V. Bhamidipati, and M. Munawer. The ARBAC97 model for role-based administartion of roles. *ACM Transactions on Information and System Security*, 2(1):105–135, Feb. 1999.

[21] C. Serban and N. Minsky. In vivo evolution of policies that govern a distributed system. In *Proc. of the IEEE International Symposium on Policies for Distributed Systems and Networks, London*, July 2009.

[22] M. Steen and J. Derric. Formalizing ODP enterprise policies. In *Proceedings of the Third Internantional Enterprise Distributed Object Computing (EDOC99) Conference*, pages 84–94. IEEE, September 1999.

[23] A. Wright and S. D.F. Sands: A service-oriented architecture for clinical decision support in a national health information network. *Journal of Biomedical Informatics*, 41:962–981, March 2008.