

# Model-Driven Policy Framework for Usage Control-based Privacy *Position Paper*

Basel Katt and Michael Hafner

University of Innsbruck, AUSTRIA  
{basel.katt, m.hafner}@uibk.ac.at

**Abstract.** In this position paper we present a framework for the modeling and enforcement of usage control-based privacy policies. In this work we introduce only PIM (Platform Independent Model) meta-models and briefly sketch how they can be transformed into the meta-models of PSM (Platform Specific Model) and finally generate the security architecture.

## 1 Introduction

Electronic healthcare systems have gained a considerable attention from researchers and stakeholders in healthcare alike. The aim is to enhance healthcare service quality. The sensitivity of the data exchanged and dealt with in such systems poses a daunting challenge to security. This is due to the fact that personal healthcare data is processed and stored on client systems as well as on the service provider's side.

Apart from traditional security requirements privacy plays an important role in this domain. Tackling the problem of privacy requires considering three problem: (1) the integrity of the client machine and its trustworthiness (2) the security(policy) model that copes with privacy related requirements and finally (3) the enforcement engine that is able to enforce these policies.

Considering privacy in the healthcare domain requires more expressive policy models than the traditional access control policies and more complex enforcement mechanisms. Let us consider the privacy requirement that states that the document must be stored on the machine of the general practitioner for only three months (this is so-called *retention time*). Another requirement is to be sure that the patient is present while the document is accessed (this is so-called *4-eyes principle*). Traditional access control policy models fail to meet these requirements: enforcement requires controlling the whole usage of the resource on the client side. Hence, usage control and obligation policy models, in which the resource (healthcare data) is monitored and controlled after being released, have to be used.

To tackle the problems mentioned above an approach was presented in previous work [6] following the PEI (Policy, Enforcement and Implementation) security engineering framework [8]. This framework distinguishes the problem of *what* security requirement is, and *how* these requirements can be satisfied or

enforced with three model layers. The policy model presented is an extension of UCON (Usage CONtrol) model [7]. In the enforcement level we proposed a general usage control enforcement model and finally introduced a proposal for usage control policy specification based on XACML.

In this position paper we propose a model-driven approach for considering security aspects in the early phases of the development cycle of healthcare related applications. We are concentrating on the creation of usage control-based privacy policies and their enforcement.

## 2 Related Work

The present work is related to and a continuation of a previous effort. In [6] we have adopted the UCON policy model for usage control and extended the model with post-obligation elements. Besides, an enforcement engine and a prototypical implementation for the healthcare domain was proposed. In this work we are targeting a framework for usage control-based privacy policy based on MDS methodology. We develop a meta-models for usage control policies and use it to generate the privacy policies and develop the enforcement engine.

MDS approach has already been used and presented in many related work. One of the most mature work in this area is the SECTET framework [5,3]. SECTET is a Model Driven Security Engineering framework for B2B inter-organizational Workflows. While we are following the same methodology, we are dealing with additional different requirements and target architecture. This stipulate considering extension meta-models to integrate the security requirement and generate the security artifacts and architecture implementation.

## 3 Privacy Requirements

Based on some practical studies in the eHealth domain [2] and studies that deal with authorization [1] and privacy [12] issues in eHealth systems, we can derive examples of privacy requirements in the healthcare domain:

- Purpose: represent the purpose, upon which the access to the healthcare document is required.
- 4-eyes principle: the presence of a patient should be checked during the access session to the record [11].
- Retention time: a patient record should be saved in the doctor's machine for a maximum one month.
- Patient consent: after the end of a treating session, the retrieved document should be stored in the local machine of the doctor in case the patient approves it; otherwise it should be deleted from the system.
- In case the patient is not present before the normal termination of a treating session, the document must be deleted and an abnormal session notification should be reported to the service provider.

The policy language to represent these requirements goes beyond the capability of traditional access control policies. That is because the patient’s record should be controlled and monitored after it was released and for the whole usage period. Hence, we use a usage control policy model to represent the privacy requirements.

## 4 Model Driven Security for Privacy Framework

Our framework is focused on healthcare systems that process and visualize healthcare data. This is due to the fact that these are the applications that access, present and manipulate sensitive health data. The applications have to be monitored by the enforcement engine. There are basically two enforcement strategies. The enforcement engine or the reference monitor of the usage control policy can be either integrated in the healthcare application or integrated as a component into the architecture to monitor and intercept all event and actions of the application. The first approach was first introduced by Scheider [9] and called *Inline Reference Monitor (IRM)*. In this approach the code of the application is analyzed and the *IRM rewriter* insert enforcement code according to the enforced policy [10]. The traditional reference monitor is located outside the application and intercept all requests to the application.

### 4.1 MDS Approach

As aforementioned in section 1 we adopt the extended UCON policy model to represent privacy policies. As suggested in [6] a usage control policy is constructed based on the states and transitions of the system. This way, continuous control can be kept over the resource. Our approach divides the usage control meta-model into two views: the *static view* and the *dynamic view*.

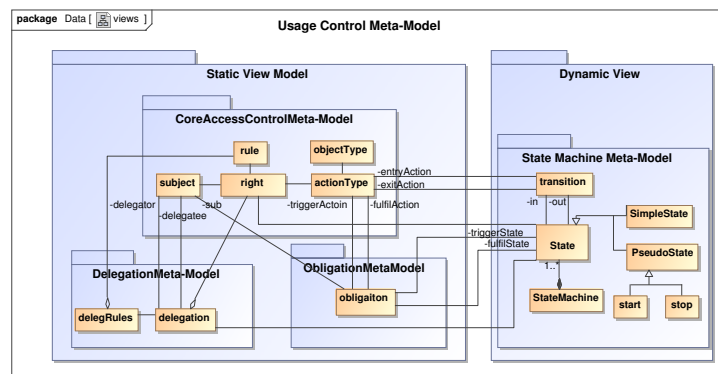
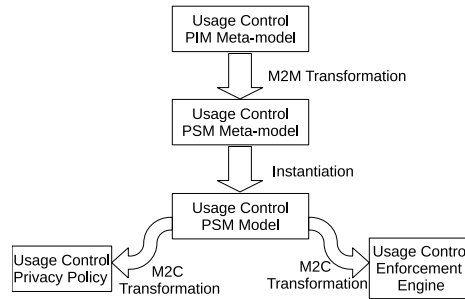


Fig. 1. Different views of the PIM metamodel

Figure 1 illustrates the different views on the PIM meta-model. The static view is divided into three meta-models that represent different authorization requirements or aspects. The main meta-model is the *Core Access Control Meta-Model*. It contains the main elements of any authorization policy. These elements represent which *rules* have to be checked to allow (*right*) a *subject* to execute an action of the type *actionType* on the resource *objectType*. This core meta-model can be extended with different aspects like delegation and obligation. For example the *Delegation Meta-model* defines the structure for modeling delegation requirements and the *Obligation Meta-model* defines the structure for modeling obligation requirements. The Obligation element for instance consists of: (1) two actions, namely the *triggerAction* that triggers the obligation policy and the *fulfilAction* that must be fulfilled, (2) two states associated with each action, and finally (3) the obligation subject. As mentioned before, privacy requirements need more than the static instance check of authorization rule. They need an on-going control of a resource’s usage. This dynamic nature of usage control or privacy policy can be represented by a statechart diagram. In such a diagram we can assign the authorization rule to be checked under specific circumstances of the system (e.g. the states). Hence, a dynamic view is added: a *State Machine Meta-model* to represents the dynamic behavior of the policy. One important point to notice here is that each state can contain different rules that must be checked when the system enters that state.

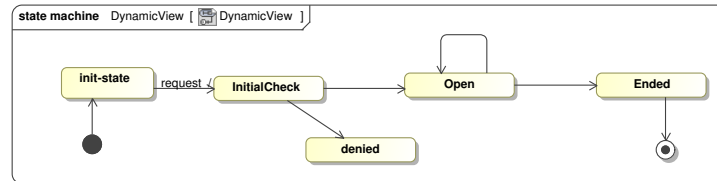
From the PIM meta-model we can generate meta-models for specific platform using Model-to-model transformation rules. For example in the context of Web service application using WS-Policy or using XACML platform. An instance of the these meta-models will represent our model (cf. Fig. 2)



**Fig. 2.** Model Driven approach for privacy related artifacts’ generation

An example of the PSM State machine model is depicted in Figure 3. It shows that upon a request to open the health record, the system goes to the *initialCheck* state, in which the rules that are assigned in this state will be checked, for example that the patient is also present. If the rule check permits the access, the record will be opened and the system will move to the *open* state. Otherwise it will move to the *denied* state where some obligation actions

can be performed and checked, e.g. the deletion of the record from the doctor's system. Please note this model is just an hypothetical example, however in a real world application it should represent the privacy requirement of that particular application.



**Fig. 3.** instance of the dynamic view PSM model ( detailed transitions are omitted for brevity)

## 5 Conclusion

In this position paper we present a model driven approach for a privacy policy framework based on usage control concept. We present the PIM meta-model and sketched a hypothetical example of a transformed PSM model (in our case it is a state machine diagram). The framework enables the generation of usage control-enabled privacy policy and helps generating (part of) the corresponding enforcement engine.

## References

1. M. Alam, M. Hafner, M. Memon, and P. Hung. Modeling and enforcing advanced access control policies in healthcare systems with sectet. *Mothis*, 2007.
2. M. Hafner, R. Mair, R. Breu, B. Agreiter, S. Unterthiner, and T. Schabetsberger. Health@net. die verteilte elektronische gesundheitsakte- eine fallstudie in modell-getriebenem security engineering. *IT-Sicherheitskongress des BSI*, 2007.
3. M. Hafner, M. Alam, and R. Breu. Towards a mof/qvt-based domain architecture for model driven security. In *MoDELS*, pages 275–290, 2006.
4. M. Hafner, R. Breu, B. Agreiter, and A. Nowak. Sectet - an extensible framework for the realization of secure inter-organizational workflows. In *WOSIS*, pages 47–57, 2006.
5. M. Hafner, R. Breu, and M. Breu. A security architecture for inter-organizational workflows: Putting security standards for web services together. In *ICEIS (3)*, pages 128–135, 2005.
6. B. Katt, X. Zhang, R. Breu, M. Hafner, and JP. Seifert. A general obligation model and continuity enhanced policy enforcement engine for usage control. *SACMAT '08: Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*, 2008.
7. J. Park and R. Sandhu. Towards usage control models: Beyond traditional access control. In *Proc. of Seventh ACM Symposium on Access Control Models and Technologies*, 2002.
8. R. Sandhu, K. Ranganathan, and X. Zhang. Secure information sharing enabled by trusted computing and pei models. In *Proc. of ACM Symposium on Information, computer and communications security*, 2006.
9. F.B. Schneider. Enforceable security policies. *ACM Trans. Inf. Syst. Secur.*, 2000.
10. U. Erlingsson and F.B. Schneider. Irm enforcement of java stack inspection. In *SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy*, 2000.
11. G. Vogt. Multiple authorization- a model and architecture for increased, practical security. In *Proc. of IFIP/IEEE Symposium on Integrated Network Management*, 2003.
12. G. Yee, L. Korba, and R. Song. Ensuring privacy for e-health services. In *Proc. of The First International Conference on Availability, Reliability and Security*, 2006.