A Security Contextualisation Framework for Digital Long-Term Preservation

Kun Qian[1], Maik Schott[1], Christian Kraetzer[1], Matthias Hemmje[2], Holger Brocks[2], Jana Dittmann[1]

[1] Faculty of Computer Science, Otto-von-Guericke University Magdeburg, Germany
{kun.qian,mschott,kraetzer,jana.dittmann}@iti.cs.uni-magdeburg.de
[2] Faculty of Mathematics and Computer Science, University of Hagen, Germany
{matthias.hemmje,holger.brocks}@fernuni-hagen.de

Abstract. Nowadays a growing amount of information not only exists in digital form but was actually born-digital. Digital long-term preservation becomes continuously important and is tackled by several international and national projects like the US National Digital Information Infrastructure and Preservation Program or the EU FP7 SHAMAN Integrated Project. The very essence of long-term preservation is the preserved data, which in turn requires an appropriate security model, which is so far often neglected in the preservation community. To address this problem, we extend the security relevant parts of the Open Archival Information System (OAIS) standard, in which security aspects are underspecified, by a conceptual framework for hierarchical security policy development based on given use-cases for a long-term archival system. The corresponding policies are then distributed and implemented by applying an iterative procedure to turn them into rules before these are then finally enforced. In this paper we describe how to construct a corresponding context model and derive such policies using the iterative approach to assure the system and data security.

Keywords: context model, digital archive, security policies, system security

1 Introduction and motivation

In this paper we perform security-oriented context modelling as well as policy generation, implementation and enforcement focused on the security of a digital long-term preservation environment which currently focuses on archiving texts (i.e. PDF files) and digitised books (i.e. TIFF files). This context model is based on the established OAIS ISO standard [1] as well as our previous work on digital long-term archival system security. In [2] we describe a use-case-centric approach of deriving operations, actions, objects, rights and roles from user-cases and how to employ these for usage within a security model, e.g. an extended version of the Clark-Wilson model [3] including a syntactic-semantic integrity and authenticity verification approach. This extended Clark-Wilson model is in [4] combined with an extended Information Lifecycle Model [5] developed within the EU FP7 SHAMAN integrated project [21]

to form a secure preservation framework for images and describing in detail the integrity and authenticity verification processes.

The work described in this paper aims at the development of a concept for implementing and managing security for digital long-term preservation environments of all kinds. The main instrument we foresee for this is the usage of policies, which of course provides the following two challenges for this paper: First, to define a suitable security-oriented context model for archival systems to act as basis for the policy generation. Second, the (security) policies themselves have to be derived from the context model. As a basis to address these two challenges we take use-cases based on the OAIS standard [1] for digital archival systems.

The main scientific contribution of this paper is the conception of a contextualisation framework for context model and policy generation as well as policy implementation and enforcement in the scope of multimedia archival systems and data security.

This paper is structured as follows: Firstly, in section 2 we present the state-of-the-art in methodologies for context modelling, policy generation, implementation and enforcement in security relevant contexts. Then in section 3 we present our concept for contextualisation and policy-based security realisation. At last in section 4 we finally conclude and summarise this work.

2 State-of-the-art

This section introduces the state-of-the-art in methodologies for context modelling, policy generation, implementation and enforcement in security relevant contexts. In the scope of IT security, a good context model reflects the characteristics, the intended application scenarios as well as corresponding threats for a system and allows the design and implementation of policy controlled security mechanisms that enforce the security aspects that are required to protect the application scenarios against the threats.

2.1 Methodologies for context modelling in IT security

Context modelling, differing from other modelling methods, not only describes the entities involved in a system but also explains how the entities are related with each other by revealing their causality and relationships. The design of a context model can either start from the very basic knowledge and be progressed by gradually adding necessary information to achieve proper complexity (“bottom-up”) or start from the vivid representation of the physical world and be progressed by gradually removing redundancy to achieve the proper simplicity (“top-down”). Thus a well-designed context model is at the same time a well balanced compromise between complexity and simplicity, always being sophisticated enough to offer all the necessary details yet still straightforward enough to be understood and applied.

In the field of IT security context modelling plays an essential role in various aspects. To meet the requirement of *confidentiality*, context modelling is for example used to implement various access control policies. For example, Bhatti et al. developed their model for web-services in [6], based on an extended, trust-enhanced

version of Role Based Access Control (RBAC) framework that incorporates context-based access control. To recognise better the broader context in which security requests arise, the Task Based Access Control (TBAC) extends the traditional subject/object based access control models by including domains that contain task-based contextual information [7]. To provide security for computing infrastructures in which access decisions may depend on the context. Covington et al. developed a context-aware access control by extending RBAC with the notion of environment roles [8]. For the aspect of *authenticity*, various models have been proposed. In the scope of sensor forensics, Fridrich describes in [20] a simplified sensor output model, which contains the basic elements of the process of digital cameras acquiring images, and then applied the model to derive a maximum likelihood estimator for the sensor fingerprint, which can be used to identify digital cameras. In [9] we propose a context model for microphone recording by describing the involved signal processing pipeline to reveal the influential factors that might be used as characteristics of different microphone to contribute to microphone authentications. More often, when a context model is developed aiming at assisting the construction of an information system, multiple aspects of security issues need to be covered. For instance, the policy model for clinical information systems developed by Anderson in [10] focuses not only on *confidentiality* and *availability* but also *integrity*. The context model described in this paper for the application scenario of a secure digital long-term preservation archive system is explained in detail in section 3. It takes not only *confidentiality*, *authenticity*, *integrity* and *availability* but also *non-repudiation* in consideration.

Currently, there exist three most prominent context modelling approaches [11]: *Object-role based context modelling* originates from attempts to create sufficiently formal models of context to support query processing and reasoning, as well as to provide modelling constructs suitable for use in software engineering tasks such as analysis and design. This approach is generally not applicable for hierarchical structured modelling. *Spatial context modelling* focuses on location information. It is well suited for context-aware applications that are mainly location-based, like many mobile information systems. *Ontology-based context modelling* exploits the representation and reasoning power to describe complex context data that cannot be described by simple languages [12]. It provides formal semantics to context data and thus makes it available to check for consistency of the set of relationships describing a context scenario as well as to recognise that a particular set of instances of basic context data and their relationships actually reveals the presence of a more abstract context characterisation. Compared to simpler approaches, it provides clear advantages in terms of expressiveness and interoperability, which is the reason we use ontology-based context modelling in our framework conception. In our concept the ontology describes the entities in the security system as well as the relationships among the entities, both of which are described by digital long-term preservation use-cases taken from the SHAMAN research project.

Some evaluation criteria on the performance of context models have been proposed in literature. For example, Strang et al. point out that the demands for context modelling include distributed composition, partial validation, richness and quality of information, (in-) completeness and ambiguity as well as level of formality and applicability to existing environments [13]. However these evaluating aspects are rather based on the requirements of context modelling for ubiquitous computing, thus

suitable metrics for context models in the scope of IT security are still to be developed. This point is not addressed here but reserved for our future work.

2.2 Methodologies for security policy generation and enforcement

Context models describe entities as well as the relationships among the entities in a system. Good policies, as the systems governing mechanisms, are the foundation of well-secured systems. Simply speaking, security policies define what in the system should be protected [14] to meet different aspects of security requirement depending on the application scenario. Baskerville's approach from [13] can be considered as a functional hierarchy of policies, using a three level division: meta-policies are "policies about policies", which declare plans for creating and maintaining security policies; high-level policies are security policies which are high-level overall plans embracing the general security goals and acceptable procedures; low-level policies are defined information security methods of action that are selected from among alternatives and applied based on given conditions that guide and determine present and future information security decisions. This functional hierarchy increases in granularity from the abstract meta-policies to specific detailed policies, which may be so concrete that they directly demand or prohibit certain implementations or mechanisms. An issue is, if abstract policies are made more specific in a parent-child interactive relationship, this will also refer to the system or, analogously the other way around, distinct parts of the system get their own low-level child policy which is a refinement of a high-level parent policy for this very part. As such, for complex system there may be large numbers of low-level child policies, and many of them may originate from a single parent high-level policy. Thus management of these can become quite complicated as changes of a policy regarding only a special system module can either only be made at a high level, which would require the revalidation of a vast amount of its child policies for every policy referring to this module. Therefore to solve this issue, in extension to Baskerville's scheme [14], we propose the introduction an additional hierarchy level between high-level and low-level policies. This new level, called mid-level policies, is intended to encompass policies that only refer to such larger system modules.

Besides the policy hierarchy considerations, for this paper we also adapt a policy life-cycle model from Baskerville et al. [14] for security focussed policies. The adapted life-cycle contains for each policy the following phases:

Specification of policy requirements: The identification and classification of security objects and subjects are two essential requirements that have to be encompassed by context modelling prior to the policy design and implementation, as the meta-policies should ensure that these requirements become primary features of the security policies. Security objects are the security relevant assets of the system, and security subjects refers to the different entities that have a relevant security connection to the objects. In the context model describing the objects and subjects, also the connections between these (e.g. access levels and types) have to be specified.

Policy design processes: In general, some form of meta-policies should specify the process by which the lower-level policies are generated and enforced. For a complex system, the usage of a hierarchy of policies ensures the required scalability. The granularity of the different security policy levels in this hierarchy should be specified

in the design. As mentioned above, we use in this paper a hierarchical policy approach with meta-policies, high-level, mid-level and low-level policies. The policy design process also includes decisions on policy expression languages and policy distribution as well as enforcement. Some policies should be enforced technically with computer technology (i.e. using access control software), some policies should be enforced organisationally, while some other policies should be enforced using personnel-focused mechanisms (like training of users or raising security awareness). Furthermore, the design of policies enforced technically should also consider the intended expression, distribution and enforcement standards (see the remarks below).

Policy implementation: How the policies are to be implemented based on expression languages and standards should also be determined and specified using meta-policies. The implementation also includes policy testing. Here functional evaluations as well as investigations on potential policy conflicts have to be performed. Nevertheless there is a usual problem that the implementation encounters: the policies are expressed in a natural language and thus too complex. In our concept this problem is solved by applying a manual and iterative procedure to turn them to enforceable rules, which are defined as formalised atomic descriptions of specific actions. More details are offered in section 3.2.

Policy enforcement: Boyle et al. developed the Common Open Policy Service (COPS) standard [15], which serves well for typical policy-based systems such as Authentication, Authorisation and Accounting (AAA) systems [16]. Using COPS the policies can be enforced via a three-tier-model: Policies are stored in the Policy Repository (PR), which could be a database, a flat file, an administrative server, or a directory server [17]. A Policy Definition Point (PDP) retrieves the policies from PR, parses and evaluates them and sends necessary commands to policy targets [18], while a Policy Enforcement Point (PEP) communicates directly with the policy targets and gives instructions of performing the policy actions following the received commands [17]. For communication between PDP and PEP, a query and response protocol is developed for exchanging policy information and decisions between them [17]. It is designed to operate reliably and in real time with minimal overhead, thus it provides a dedicated QoS controller for the PEP. Additionally, when necessary, Local Policy Decision Point (LPDP) can be defined between PDP and PEPs. In this case the PEPs take policy decisions from the LPDP for their domain, while the PDP remains the authoritative decision point at all times. Parallel to the enforcement an auditing of the system has to be performed where some monitoring mechanism should detect any failed enforcement attempt or policy conflict. In this enforcement phase also the execution of replacement or termination of policies is performed.

3 Design of our contextualisation framework

Based on the state-of-the-art presented in section 2, here we describe our framework for contextualisation of security for digital long-term preservation. This framework consists of four major functional blocks: *context modelling*, *policy generation hierarchy* with its different stages, *Information Package (IP) processing* and *control*. The *context modelling* block consists of two distinct parts: global (system-wide) and local context modelling. The *policy generation hierarchy* is a hierarchy of stages

beginning at the top with the generation of system-wide global policies and ending in the deviation and invocation of rules for IP processing operations. The *IP processing* itself identifies the IPs (or related system data) to be processed and applies the rules as sequences of atomic data processing operations. The *control* block controls the *context modelling* and the *policy generation hierarchy* and acts as a central policy repository as well as a central audit service for the overall system.

Within the following sections we describe these functional blocks in more detail and show how to model context, generate and implement as well as enforce security policies based on use-cases from a data intensive, complex, security-oriented data processing system like an archive for digital long-term preservation.

3.1 Context modelling for complex, security-oriented data processing systems

A complex data processing system usually contains multiple processing entities with different types of relationships among the entities. Therefore the "top-down" modelling approach is not suitable, as achieving a complete and vivid representation as a starting point in context modelling is not feasible under these circumstances. Instead, it is more appropriate to first extract typical tasks (workflows) from use-cases in such systems and then gradually extend these into a fully developed context model. As ontology based context modelling has its speciality in organising complex structured context data, it is reasonable to apply it in the construction of the model for such systems. The resulting ontology describes the entities in the system as well as their relationships. The latter are expressed here in the form of policies and rules.

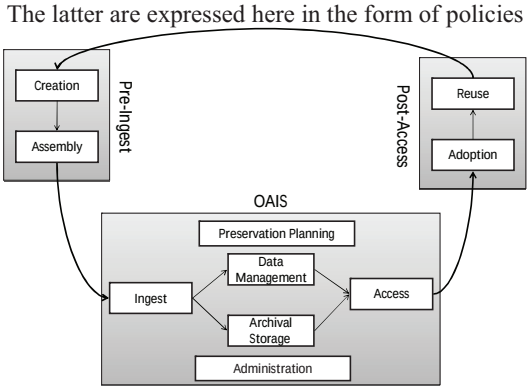

Figure 1. Phases and processes in the Information Lifecycle Model (based on [5])

In digital long-term preservation the basis for context modelling is the OAIS standard [1]. Its functional model describes several processes of an archival system, their tasks and their relationships as well as data items – thus providing a general context of systems for this application scenario (see Figure 1). In these processes typical use-cases are grouped. In the *ingest* data objects that should be preserved are received from a producer and converted into the archives data format. The *archival storage* stores and manages these data objects inside the archival system. The *data management* provides services for the discovery, access to the metadata, and maintaining the referential integrity between data objects. The *administration* process is responsible for the operation of the archival storage, procuring and installing new

hardware and software and the organisational enforcement of the policies and standards. The *preservation planning* ensures that the archival storage can fulfil its requirements by observing the technical state-of-the-art and legal requirements and adapting its policies with this regard. *Access*, as the last major process, provides services for consumers to locate and retrieve data objects or information about them.

Within the SHAMAN project Brocks et al. [5] extended the OAIS model by introducing an extended *Information Lifecycle Model*. In this the aforementioned processes from ingest to access are seen as phases of the lifetime of a data object. The information lifecycle model extends this by including the objects "life" before and after its management within an archival. The phase before a digital object enters an archival system is called "Pre-Ingest". This is further divided into the processes of the actual *creation* of the data later to be ingested and its *assembly* into a package supported by the archive. The phase after a digital object leaves an archival system is called "Post-Access" and is also divided into two processes: *adoption* where the received data is unpacked, examined, transformed, displayed or in short all tasks that are needed for repurposing the content and *reuse* where the content is actually exploited. Reuse may also include the re-ingest of this object or a derivation thereof into an archival system, leading to a real life-cycle as shown in Figure 1. Such connection of reuse and creation is especially the case for collaborative environments.

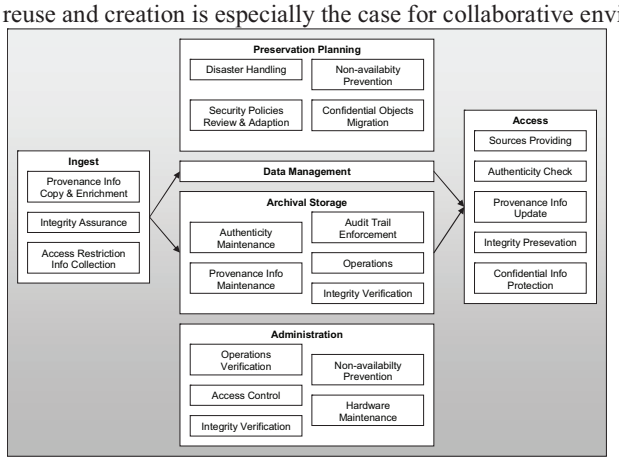

Figure 2. Required extension of the OAIS processes from a security point of view

Within this paper the considerations are limited on the central phase of the Information Lifecycle Model, the archival phases described by OAIS, and security considerations. As the original OAIS ISO standard is lacking detailed information about security requirements, it needs to be enhanced in this regard within this paper. Thus we analyse the archival use-cases provided by the SHAMAN research project and extract the tasks which would have to be considered in addition to the already existing OAIS functional entities. Thereby these extensions, which may not be separate entities but can be incorporated into existing ones, provide better context representation in terms of security. Figure 2 shows just these new tasks focused on risk mitigation for the OAIS processes, whereas the tasks and functional entities of the original OAIS model are omitted in this figure for the sake of clarity. The interested reader may refer to the OAIS documentation [1] for details on these

original tasks and functional entities.

The global context model visualised on different levels of detail in Figure 1 (Information Lifecycle Model) and Figure 2 (detailed overview over the security tasks in the OAIS processes) is then used as starting point for policy generation described in section 3.2. Each process (Ingest, Access, etc.) has its own entity taking responsibility of the local context modelling as well as the generation, distribution, implementation and enforcement of policies within its own domain (scope of the process). All entities within this domain can act as the enforcement points for policies.

3.2 Use-case-driven policy generation, implementation and enforcement

Based on the assumption that a "top-down" modelling approach is not suitable, which is explained in section 3.1, it is reasonable to derive policies for complex data processing systems from use-cases. Furthermore, to more vividly represent the complex relationships among the entities in such systems and to implement means of governance or orchestration a hierarchical organisation of the policies is applied.

As mentioned in section 2.2, in this paper we use the three-level approach from Baskerville et al. [14] and enhance it to a four-level policy hierarchy.

Our proposed policy generation starts on a global system level with the most abstract types of policies – meta-policies and high-level policies. The first makes statements about other policies and the second about general security goals and acceptable procedures on a system-wide perspective. Thus they can either be derived from use-cases making policy assertions and from system use-cases, respectively, or come from the general understanding of the system or the application scenario.

Inspired by practice of defining optional LPDPs in the COPS standard [15], we decide to add another layer of mid-level policies in the previous three-layer policy model introduced in [14], for better handling of larger complex system modules. This reflects the fact that many use-cases do not make assertions about the system as a whole, but about certain functionalities. Such use-cases are restricted to larger system modules (in our case equivalent to the OAIS processes) and their domain of functional entities. In the policy generation hierarchy these mid-level policies on the one hand serve as a process-based filter for the use-cases of which a system may have a large amount of, and on the other hand they serve to verify if the high-level policies themselves make sense by not contradicting the existing use-cases (i.e. verify the consistency between global and local context modelling).

The mid-level policies are used to act as the basis for the generation of low-level policies, which provide sufficient information what should be implemented as a rule in the enforcing. In the ideal case these low-level policies should be precise enough to directly derive rules in a formalised language from them.

For the sake of clarity and for the sake of the traceability of the policies origins, a policy derived from a higher policy should have an identifier indicating its parent policies. If high-level policies have an identifier of the format P*x* (with *x* being an unique identifier) their children mid-level policies should have an identifier that includes their parent's identifier (e.g. P*x*-*y*). As policies need to be updated or even removed at certain times, this form of traceability eases the browsing of the hierarchical tree structure of the policies that would be required in these cases.

For highly complex systems there arise some issues for the implementation and

enforcement of policies: First, when introducing a new policy into such systems, there could be multiple possible methods to implement it, thus it requires specific analyses (e.g. complexity-based) to identify the optimal method. If the COPS standard for policy management were applied in this case, these considerations would have to be also extended to policy decisions on the selection of PEPs. Second, complex systems are with a high probability also heterogeneous, therefore considerations have to be included on the interoperability, distribution and orchestration of policies and policy descriptions (for instance how to interpret between possible different policy syntaxes used in different parts of a heterogeneous system). Third, due to the quantity and complexity of the policies, it is necessary to develop an assurance and auditing mechanism to make sure that all the policies are enforced properly.

The policies considered here are basically descriptions in natural language of what the preservation system does, which creates barriers for actual enforcement. Thus in our concept, the generated policies are implemented by applying a manual and iterative procedure which turns low-level policies into enforceable rules. The procedure is described as follows:

Create Rules: This turns low-level policies, which define what needs to be done, into rules, which define how the policy is enforced. It analyses the statement in the policy by utilising validation criteria that consist for the significant properties, format validation, organisational- and domain information. Then a sequence of steps is derived, describing specific actions. Each step should be as atomic as possible, ideally performing one action and also verifiable, so it can be considered as one abstract rule. Optionally a rule can comprise sub-rules if one of the steps is too complex to be described as a single action. Therefore the output here is a sequence of abstract rules.

Instantiate Rules: Abstract rules are not executable as they only describe actions in natural language. Therefore it is necessary to derive executable rules from abstract ones. Templates containing the grammar and syntax for rule-engines can be used by a rule instantiation tool to create realisations of the abstract rules. Such tool should also keep track of the realisation process so that it is possible to track from an executable rule back to the abstract rule and then back to the policy. Additionally, similar to Event-Condition-Action (ECA) rules which always have the form of *if…then…else*, the executable rules are formalised as Semantic Web Rule Language (SWRL) [19] rules embedded in the Web Ontology Language (OWL) context representation, thus each rule becomes an executable atomic data processing operation.

Validate Rules: Here it is ensured by validation that the instantiated executable rules are correct implementations of the policies. The functionality of the used validation tools would be defined by the validation criteria, which are the adherence to the global and local context models (developed in 3.1). After a rule passed the validation, it is deployed with records of its deployment time and intended deployment enforcement point in the production system and ready to be enforced.

Once the policies are implemented, i.e. turned into formalised and validated rules describing executable actions, it is easy to enforce them. The COPS standard can be adapted to fit this case. Instead of PR, a Rule Repository (RR) would be used to store the rules. Similar to PDP, Rule Decision Point (RDP) would retrieve the rules from the RR, parse and evaluate them, then send rule decisions to rule targets, which can be either devices or humans to perform the actions. Similar to PEPs, Rule Enforcement Points (REPs) make direct communication with the rule targets and give them

instructions on performing the actions following the commands. Depending on the complexity of the system, Local Rule Decision Point (LRDP) can share the responsibility with RDP by feeding REP with detailed rule decisions, while RDP remains authoritative rule point at all times.

3.3 IP processing and Control

In the *IP processing* block a system entity (here equivalent to a rule target) enforces rules on IP from the archival system and/or system data (like search indexes, the user database, etc.). The result of the enforcement has to be communicated by the responsible REP to the central audit service. This central audit is part of the functionality of the *control* block. Besides this audit functionality there are also mechanisms for the storage of the policy tree (all policies are communicated to this repository during the construction of the *policy generation hierarchy*) as well as the policy conflict analysis and conflict resolve. The corresponding OAIS authority responsible for these operations would be the task "Security Policies Review & Adaption" in the process of "Preservation Planning" (see Figure 2). It should keep track of all the policies to ensure they operate properly, especially no policy from one phase conflicts with those from other ones, similar to the responsibility shouldered by policy decision points in the COPS standard.

3.4 Combination of the functional blocks of the framework

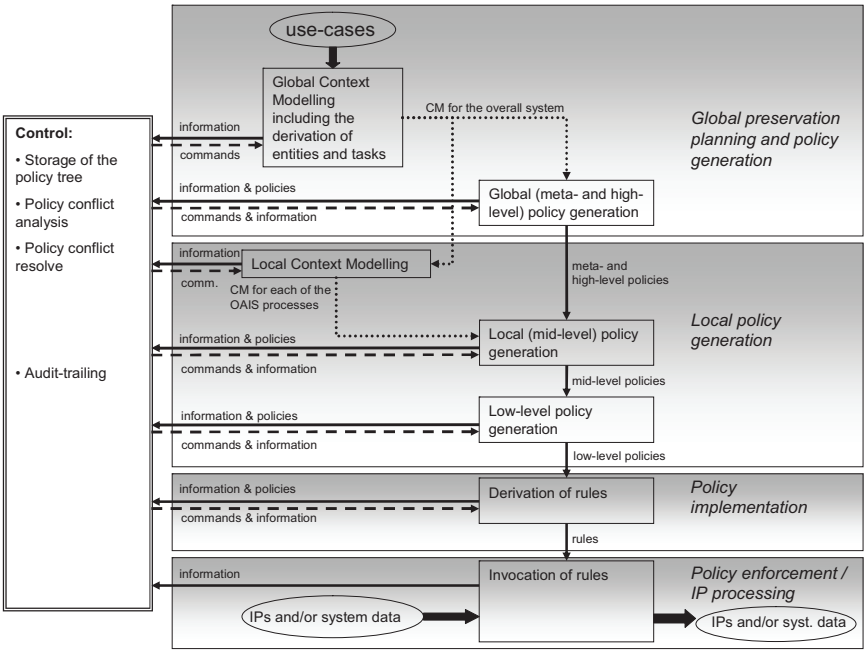

Figure 3. General overview over the security contextualisation framework

Figure 3 extends the low detail description of the contextualisation framework explained in introduction of section 3 by the data, information and control flows described for the four functional blocks in sections 3.1 to 3.3.

In Figure 3 especially the importance of the control block sticks out as a dominant factor. Each context modelling block, the different stages of the policy generation hierarchy and the IP processing communicate their actions to the control block. This is on one hand done to audit all operations for purposes of transaction control and non-repudiation of transactions as. On the other hand this functional block also acts as central policy storage repository and performs policy conflict analysis and resolve.

4 Summary and conclusions

In this paper we have outlined a bottom-up context modelling approach which derives a hierarchical policy structure from given use-cases for a long-term archiving system. An existing concept from literature has been extended accordingly into a complete contextualisation framework to meet the (security) requirements of a digital long-term preservation system.

However, there exist several limitations for our approach that have to be addressed in future work: First, it is hard to evaluate whether the constructed context model (as basis for the policy generation process) is too specific with unnecessary redundancy or too abstract with lack of necessary details, as currently no metrics for the preciseness of such models exist. Furthermore, it is difficult to investigate how vividly the lower level policies reflect the intentions of the high level policies from which they derived, yet the biases (or even conflicts) between could lead to problems in their enforcement.

Acknowledgement

The work in this paper has been supported in part by the European Commission through the FP7 ICT Programme under Contract FP7-ICT-216736 SHAMAN. The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

References

1. Consultative Committee for Space Data Systems (CCSDS): Reference Model for an Open Archival Information System (OAIS). Recommendation for Space Data System Standards, CCSDS 650.0-B-1, Blue Book (ISO 14721:2003), 2002.
2. M. Schott, C. Kraetzer, J. Dittmann, C. Vielhauer: Extending the Clark-Wilson Security Model for Digital Long-Term Preservation Use-cases, Proc. of Multimedia on Mobile Devices, 2010, SPIE Electronic Imaging Conference 7542, 2010.
3. D. D. Clark, D. R. Wilson: A Comparison of Commercial and Military Computer Security Policies, IEEE Symposium on Security and Privacy, 1987.
4. M. Schott, C. Kraetzer, N. Specht, J. Dittmann, C. Vielhauer, Ensuring Integrity and

Authenticity for Images in Digital Long-Term Preservation, Proc. of Optics, Photonics and Digital Technologies for Multimedia Applications, SPIE Photonics Europe, 2010.
5. H. Brocks, A. Kranstedt, G. Jäschke, M. Hemmje: Modeling Context for Digital Preservation, Studies in Computational Intelligence, vol. 260, pp. 197-226, 2010.
6. R. Bhatti, E. Bertino, A. Ghafoor, A Trust-based Context-Aware Access Control Model for Web-Services, Proc. of the IEEE International Conferences on Web Services, 2004.
7. W. Tolone, G. Ahn, T. Pai, S. Hong, Access Control in Collaborative Systems, ACM Computing Survays, Vol. 37, March 2005.
8. M. Covington, W. Long, S. Srinivasan, A. Dey, M. Ahamad, G. D. Abowd, Securing Context-Aware Applications Using Environment Roles, ACM Symposium on Access Control Model and Technology, ACM, Chantilly, VA, USA, 2011.
9. C. Kraetzer, K. Qian, M. Schott, J. Dittmann, A Context Model for Microphone Forensics and its Application in Evaluations, Proc. of Media Watermarking, Security and Forensics XIII, IS&T/SPIE Electronic Imaging Conference7880, San Francisco, CA, USA, 2011.
10. R. J. Anderson, A Security Polity Model for Clinical Information Systems, Proc. of IEEE Symposium on Security and Privacy, 1996.
11. C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, A Survey of Context Modelling and Reasoning Techniques, Pervasive and Mobile Computing, Elsevier, 2010.
12. G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, L. Tran, Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0, W3C Recommendations, W3C, 2004.
13. T. Strang, C. Linnhoff-Popien, A Context Modeling Survey, Proc. of the First International Workshop on Advanced Context Modelling, Reasoning and Management, in conjunction with UbiComp 2004, Nottingham, England, 2004.
14. R. Baskerville, M. Siponen, An Information Security Meta-policy for Emergent Organizations, Logistics Information Management, Volume 15, Number 5/6, 2002.
15. J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, The COPS (Common Open Policy Service) Protocol, RFC2748, 2000.
16. C. Rensing, M. Karsten, R. Stiller, AAA: A Survey and a Policy-Based Architecture and Framework, IEEE Network, Vol. 16, 2002.
17. R. Rajan, D. Verma, S. Kamat, E. Felstaine, S. Herzog, A Policy Framework for Integrated and Differentiated Services in the Internet, IEEE Network, Vol. 13, 1999.
18. K. Yang, A. Galis, C. Todd, Policy-Based Active Grid Management Architecture, Proc. of 10th IEEE International Conference on Networks, 2002.
19. I. Horrocks, P. F. Petal-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, Member submission 21 May 2004, W3C, 2004.
20. J. Fridrich, Digital Image Forensic Using Sensor Noise, IEEE Signal Processing Magazine, vol. 26, no. 2, 2009.
21. SHAMAN website: http://www.shaman-ip.eu