A Description Model for Policy Conflicts for Managing Access to Health Information

Raik Kuhlisch

Fraunhofer Institute for Open Communication Systems FOKUS,
Kaiserin-Augusta-Allee 31,
10589 Berlin, Germany
raik.kuhlisch@fokus.fraunhofer.de

Abstract. For a better business and IT alignment in hospitals an ontology-based description model for policy conflicts is introduced. Such a model is a necessary prerequisite for the subsequently domain-specific policy conflict handling as a hospital information management related activity.

Keywords: E-health, hospital intra-enterprise policy conflict, policy compliance verification, information security, conflict model

1 Introduction

One of the key aspects with regard to the hospital information management is the fulfillment of legal and regulatory requirements. This covers both patients' privacy instructions on the use and/or disclose of his data and the operational need by the (medical) staff to access classified information in order to treat them well. As a consequence the legal processing of health information needs to be directed accordingly. Policies are a feasible means to describe and regulate reasonable actions that underpin the compliance of a hospital and its staff. A recent literature review emerges that the modeling, measurement and evolution phases of business and information technology (IT) alignment is still an issue [1]. One of the main technical challenges in an alignment approach is how to define and apply such policies as means to translate organizational requirements into guidelines and rules in IT management.

There are multiple sources (e.g., legal provisions, regulations, patient consents) given that limit the processing of medical data and might be expressed with various kinds of policies. Briefly, major policy concerns for healthcare information exchange that can be applied in hospitals are (1) patient privacy consents, (2) purpose of use, and (3) compliance [2]. The patient explicitly identifies authorized identities that are allowed to use his data. The consent might have additional prerequisites such as an objection to the use of pre-treatment data. Derived constraints from the intended use of a certain hospital information subsystem that mediates access to protected information are subsumed under the term purpose of use. This describes limitations of the overall functionality of a subsystem under specified conditions (e.g., certain

medical treatment step or workflow). It provides information on which tasks can be carried out using the specific application systems and what actions on the underlying data are permitted. The information management compliance concludes the policy concerns. Since the hospital is liable for the lawful processing of the patient data, compliance allocates responsibilities (roles, permissions, and obligations) for internal and external communication of data.

In support of the above mentioned monitoring activities it should be investigated to what extent different levels of policies during regular operation are compliant to each other. This paper addresses conflicts between these policy concerns as well as concrete policy types. As a first contribution, an ontology based description model for policy conflicts that might melt the protection of patient privacy is presented.

The remainder of this paper is structured as follows: Section 2 introduces specific policy types derived from generic policy concerns as a background for policy conflicts. In Section 3, our conflict model is defined. Section 4 briefly reflects a methodology for detecting policy conflicts. Section 5 summarizes this paper and future research directions are addressed.

2 Policies for Accessing Patient Data

The general notion of a policy is to regulate the system behavior: Instead of re-implementing parts of the system solely the respective policy is applied [3]. Taking for granted the ideas of a definition of a security policy in [4], *organizational security policies* and *automated security policies* can be differentiated in general. An organizational security policy is understood as a set of rules and practices that determine how an organization manages and protects its resources to achieve defined security policy objectives. Quality criteria, such as accuracy, reliability and robustness of a system are excluded from this definition since this cannot be directly protected. Restrictions and properties that determine how an IT system prevents access to information and its resources are regarded as automated security policies [4].

These security policies can be applied to the information management in hospitals, too. Obviously, policy hierarchies with the range of corporate policies, task oriented policies, functional policies, and low-level policies can be found [3]. In addition, there are other factors which are relevant for protecting health information or patient data.

The shell model in Figure 1 depicts influencing policy factors for medical data access which are previously investigated in [2]: The ultimate baseline is the national law which scopes the legal disclosure of patient data by members of health care professions. This is accomplished by a patient consent to shape the processing of patient's data. Due to the contractual nature of the consent the following forms are to be distinguished: implied consent, presumed consent, and explicit consent. The first consent type may be caused by the appearance of the patient without his written consent (his declaration of intent is assumed). The second type may be caused when the patient is unable to express his consent to the disclosure of his personal data, but he would reasonably do (e.g., in an emergency case). The latter one may be made orally. However, for reasons of securing evidence this should be in writing.

Nevertheless, patient privacy consents join the policy factors for medical data access. The corporate governance strategy defines compliance requirements—even for information management. Corporate governance defines IT strategies with roles and responsibilities and makes requirements for the organization of work. Finally, application systems mediate the access to patient data that is governed by an identity and access management.

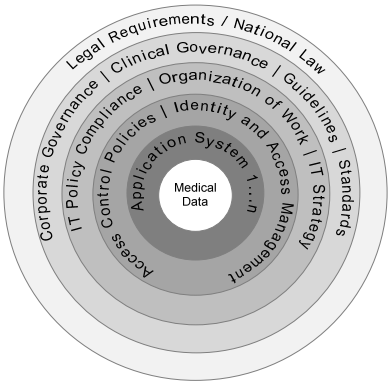

Fig. 1. Influencing policy factors for medical data access

2.1 Policy Types

Based on these factors, the security policy definitions above and definitions in [2] the following different policies types can be defined which is shown in Figure 2.

- An *information access policy* regulates who is authorized to disclose information by defining of confidentiality of at least one specific classified information object. Therefore, access demands to information result. Such policy is a special part of an overlying *information security policy* that in turn contains further regulations and information such as an overview of the corporate philosophy on security, a statement of purpose, and organization's security responsibilities that define the security organization structure [5, p. 248].
- A *resource access policy* is the special part of a *patient privacy policy* that authorizes certain individuals and organizations to use the HIS subsystem with regard to the agreed purpose of use. Technically, safeguarded entry points are authorized by this policy. A policy template might be *"I hereby authorize [roles] at [organizations] to use the 'Historical Database' Application in order to access all [Patient] [kind-of-data] for the purpose of [purpose]"* which may result in the following instance: *"I hereby authorize physicians at Clinic A to use the 'Historical Database' Application in order to access all my lab data for the purpose of medical treatment"* [2].
- A *resource behavior policy* is the counterpart of the resource access policy since it defines how certain subjects might act on certain HIS subsystems based on

their functional/structural role. Whereas the resource access policy is patient-driven, the resource behavior policy is organization-driven (i.e., it reflects the structural organization and process organization).

– An *access control policy* contains the actual access rights configured in a dedicated HIS subsystem. There are several nuances of this policy type defined in [6]. Hence, *authorization policies*, *obligation policies*, *refrain policies*, and *delegation policies* are to be distinguished additionally.

This overview reveals the versatility of policies in hospitals. To conclude, an information access policy and a resource access policy are regarded as 'pure' organizational security policies with respect to the above definitions. On the contrary, a resource behavior policy and an access control policy complement the regulated access to patient data as automated security policies.

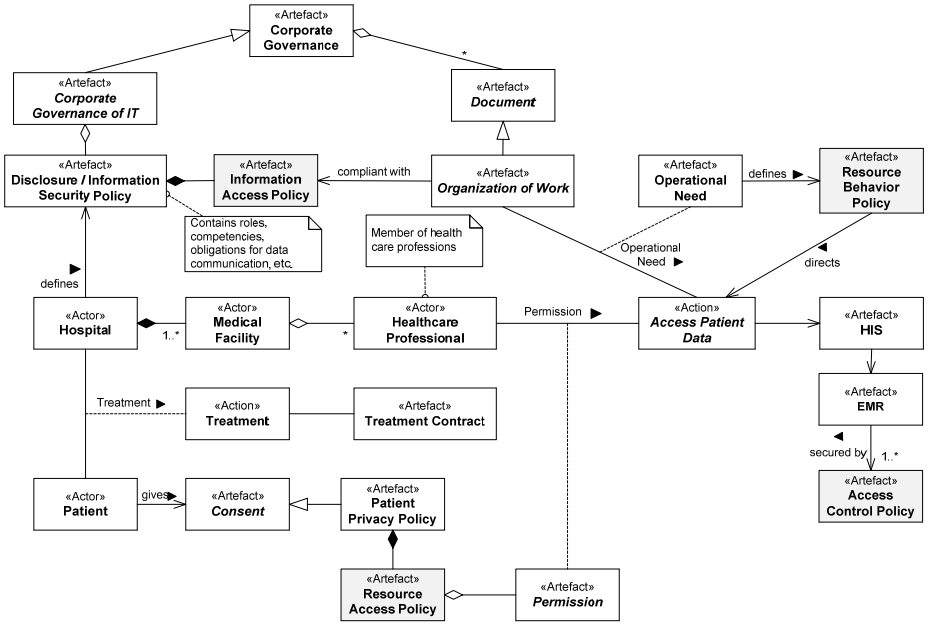

Fig. 2. Range of policies to control the patient's data access (UML)

3 Formalization of Policy Conflicts

Policy conflicts might occur between policies of the same type/concern (e.g., two rules of two access control policies may contradict each other) or crosscutting (e.g., a patient disagrees with the usage of a health care application in a resource access policy but is a necessity for the organization of work that is reflected by a resource behavior policy). The challenge is to identify conflicts between different policy concerns since more factors for data access have to be considered. This section

defines policy conflicts. Afterwards, the development of the description model is introduced.

3.1 Policy Conflict Notion

It is obvious that similar elements (e.g., subjects, actions) must be included in the various policies to give rise to a conflict. According to [7], a policy conflict "occurs when the actions of two rules (that are both satisfied simultaneously) contradict each other". This definition is vague since it does not look at the different policy concerns. That is why policy conflicts are either of the same policy type or crosscutting. The latter one describes contradicting contexts and intexts (i.e., policy structures and behavior).

3.2 Policy Conflict Types

On the basis of the conflicting concerns defined in [2] the following policy types might be in conflict:

- *Conflicts between information access policy and resource behavior policy.* Likely, a common occurrence e.g. when access to application systems is configured ad-hoc without raising the question such as "why do an attending physician has access to the patient administration system".
- *Conflicts between information access policy and access control policy.* This can solely happen if resource behavior policies are not synced with access control policies. That is, if entry points for application systems are safeguarded correctly, unknown identities cannot be authorized in an authorization policy. However, access might be denied although someone is allowed to use a dedicated application system (e.g., due to time constraints).
- *Conflicts between resource access policy and resource behavior policy.* Another likely scenario where a patient might restrict the use of his data for a special purpose of use, but a physician is authorized to use an application system to his task (e.g., hygiene or quality management).
- *Conflicts between resource access policy and information access policy.* Such conflicts arise when patients restrict/expand their use of data which simultaneously is a compliance breach. This is the case when identities are granted to access patient data via dedicated application systems but the organization of work does not consider such accesses. However, under the assumption that the resource behavior policy correctly implements the defaults by the information management (i.e., compatible with an information access policy) no patient data breach can occur.
- *Conflicts between multiple resource access policies.* Different permissions from patient privacy policies refer to different entry points to application or application systems. This conflict occurs if more than one resource access policies with the same context (subjects, resource) are activated for patient and have different actions are authorized.

- *Conflicts between multiple access control policies.* Actions or rather access rights contradict each other. To avoid such conflicts (activated) policies must be considered in the total. However, if the same subject is granted access to data via a specific application/application system and in turn denied to access the same data via another application/application system represents no conflict. Hence, different purposes of use are implemented.

From these conflicts the following classifications are extracted:

- *Positive-negative conflict of modalities:* By analogy with the structured model of policies and its conflict analysis defined by Moffett et al. [8], modality conflicts are conflicts regarding (to be authorized) actions and their associated policy goal. That is, on the one hand a modality expresses whether specific actions must be initiated or prevented (these actions refer to obligation and refrain policies) to achieve a goal—on the other hand it occurs if the execution of specific actions is permitted or forbidden (this refers to authorization policies). For instance, if one policy grants and another one denies access to the same resource it is a modality conflict.
- *Functional dependency:* The policy execution order is important for so that a conflict occurs. Or one policy requires permissions from another one (e.g., in delegation scenarios). This conflict type indirectly represents a priority conflict of access rights.
- *Term or attribute conflicts:* Especially when role-based access control scenarios are implemented, this conflict might occur. For instance, roles or attribute groups might have different meanings in other application domains and thus might create conflicts [9].
- *Semantic mismatches between policy concerns:* This conflict refers to goal conflicts (e.g., restriction of information implemented different in policies). Since actions implement/achieve the goal there might be a contradiction. For instance, the purpose of use in a resource access policy and a resource behavior policy have nothing in common. Hence, if the semantic match (i.e., similar concepts) cannot be checked, a conflict arises.

3.3 Modeling

Enterprise modeling serves as a useful tool for business and IT alignment [10]. This even applies to policy conflict analysis. Ideally, valid policies (i.e., conflict-free policies) are derived from one or more information models and deployed to application systems. Thus, a model has a verification-validation-testing function [11], which is essential for policy planners. The conflict model is based on a platform-independent model as an ontology. In the terminology of the model-driven development strategy MDA this model can be described as platform-independent. The approach is to first formalize the policy types and then define the concept of conflicts.

Choice of Language. Semantic Web technologies provide a rich pool of techniques for representing structured knowledge from weak to strong using ontologies.

Moreover, its logical reasoning and treatment of new contexts seem to fit very well to handle different policy types. Promising candidates are the Resource Description Framework (RDF) Schema[1] and the Web Ontology Language[2] (OWL) for structuring RDF resources. RDF Schema and OWL share the same syntax RDF/XML. With both vocabularies the possibility to define formal design models is given. When comparing the expressivity of RDF Schema to OWL, it is obvious that OWL is much more powerful than RDF Schema since its vocabulary is more comprehensive. Thus, for example, no disjoint classes are supported by RDF Schema or the necessary expressivity to define cardinalities does not exist. Nevertheless, with the primitives *rdfs:Class* and *rdf:Property* might be achieved any expressivity: Missing constructs can be reproduced by means of rules. The derivation of new knowledge (reasoning) for large ontologies with RDF Schema is—due to the lower computational complexity—easier. Since every OWL ontology is compatible with RDF Schema, it plays rather a subordinate role which language is used here. OWL provides unnecessary semantics (e.g., equivalent classes, complex classes, same individual) so that RDF Schema is used for modeling the policy types and their conflicts.

Cardinality Constraints. The expressivity of RDF Schema is sufficient. However, one handicap is that RDF Schema cannot express cardinality constraints. One workaround is to subclass *rdf:Property* named *RestrictedProperty*. This indirection gets the two *rdf:Property* elements *minCardinality* and *maxCardinality* with the range of values for integer data types. In this way, subclasses of policy objects can record cardinalities.

```
<rdfs:Class rdf:about="urn:policy-ns:basic-policy#RestrictedProperty">
   <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
   ns#Property"/>
</rdfs:Class>
<rdf:Property rdf:about="urn:policy-ns:basic-policy#minCardinality">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
   <rdfs:domain rdf:resource="urn:policy-ns:basic-policy#RestrictedProperty"/>
</rdf:Property>
<rdf:Property rdf:about="urn:policy-ns:basic-policy#maxCardinality">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
   <rdfs:domain rdf:resource="urn:policy-ns:basic-policy#RestrictedProperty"/>
</rdf:Property>
```

In order to check the integrity of cardinality with instances RDF Schema rules need to be extended (for the sake of clarity, the RDF/N3 sample solely checks whether minimum cardinality is set):

[1] See http://www.w3.org/TR/rdf-schema/.
[2] See http://www.w3.org/TR/owl2-overview/.

```
[restriction_rule_1:
      (?v rb:validation on()) -> [restriction_rule_1: (?x rb:violation
      error('ERROR', 'Cardinality Violation', ?y) ) <-
      (?x rdf:type p:RestrictedProperty), noValue(?x p:minCardinality ?y)
]]
```

Ontology Overview. The basic policy ontology defines the minimal objects of each policy: domain, goal, target, rule (with an associated action), and trigger. Since each domain (*information management*, *patient domain*, *application systems domain*, *application domain* etc.) has its specific requirements and additional *rdf:Property* elements, sub-policy ontologies are created. The sub-ontologies import the basic ontology and inherit all properties from the basic policy.

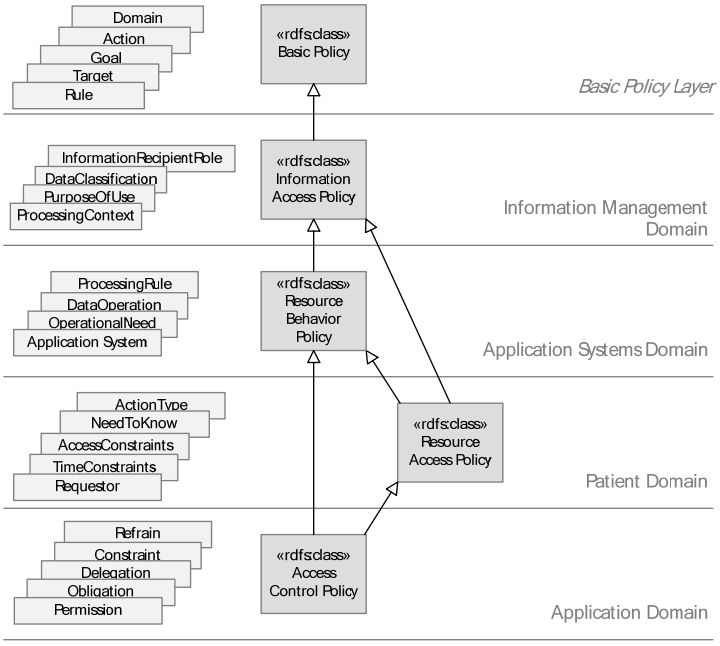

Fig. 3. Structure of ontologies

Definition of Conflicts. Each identified conflict has to be defined accordingly. For reasons of space a sample definition of a modality conflict between a resource access policy and an authorization policy is given below.

```
function modality-conflict-1 (document rap, document authzp)
   get authorized action and consent type from rap
   get configured action and rules from authp
   while authzp has more rules
      input the next rule
      get authorization decision from rule
      If consent type equals "grant access" and
         authorization decision equals "permit"
         If configured action is not a subclass of authorized action
            print "Policies do not grant access equally.";
         endif
      endif
   endwhile
end function
```

Furthermore, conflicts are defined as rules which can be applied independently to each policy type and processed by a rules engine or reasoned that ensure the proper use of the resources defined in it. Such rules can be written e.g. in RDF/N3. The sample below selects attributes from two given policies and states that the associated actions have to be in the same hierarchy.

```
{
   ?p1 a rap:ResourceAccessPolicy.
   ?p1 basic:pAction ?a1.
   ?p1 rap:consent-type "grant access".

   ?p2 a acp:AuthorizationPolicy.
   ?p2 basic:rule ?r.

   ?r acp:AuthzDecision ?d.
   ?r basic:pAction ?a2
}  => { ?a2 a ?a1 }.
```

4 A Methodology for Policy Conflict Detection

The detection of policy conflicts depends whether all relevant policy information is present. So the first step is to gather all information with regard to information management, need-to-know, applications/application systems and feasible patient privacy constraints. Supposing that a proper work organization with a user and identification management is established, such effort has to be done only once initially. The following steps might be useful to detect incorrect policy statements.

1. *Represent written and existing policies as ontology.* Tailor the given policy ontologies to the existing hospital environment. Ideally, the LDAP-based user directory can be imported or queried directly [12]. The description model uses the proposed privacy-friendly recommendation by the national data protection officers

for the design and operation of hospital information systems in Germany [13]. This can be used as a basis.

2. *Instantiate concrete policies of the ontologies.* Existing regulations must then be formalized by instantiating policies.
3. *Apply rules independently to concrete policy instances.* In order to indicate absence of conflicts rules are applied. Thus, emerging conflicting policies may be refined accordingly.
4. *Deploy policy instances to application systems.* Valid policy instances are transferred into policy languages that are supported by the application systems such as XACML [14]. Moreover, the representation of privacy consents as Clinical Document Architecture (CDA) Release 2 (R2) documents in accordance with the upcoming normative Health Level Seven (HL7) standard[3] is advised.

5 Related work

Policy conflict handling is investigated comprehensively. For instance, Kempter et al. [15] propose the use of models to support conflict handling. They map invariants (rules/dependencies from a managed system) to policy actions. Conflict definitions are derived from the invariants. Conversely, they do not look at semantic conflicts as they occur in health care related policies environments.

In addition, Aphale et al. [16] give guidance for logical and functional conflict identification and resolution. Their work is based on activities which are ranked through a prioritization model by means of heuristic mechanisms in order to achieve an individual or organizational goal. Logical conflicts refer to the before mentioned conflicts of modality whereas functional conflicts describe inconsistent pre-conditions between actions and a goal of the organization (i.e., need-to-know). However, the developed agent assistant based on OWL 2.0 focusses on goals of an organization and does not regard different policy concerns such as patient constraints.

The detection of conflicts across different policy concerns is investigated in [17]. This is a similar approach since conflicts are based on a domain description model and class-specific policy models. Conflicts are expressed as rules. The approach in this work differs because it considers the information management in a top-down manner and captures the specifics of each policy class. Conflicts in [17] are not classified but referred to possible impacts or threats which should be embedded in a security risk management point of view.

6 Summary and Future Work

This paper proposed the use of a semantic model for capturing the specifics of relevant policies in a health care environment (especially in hospitals). It is useful to detect policy conflicts when authoring new policy statements. Different policy concerns are represented as dedicated policy types via separate ontologies. These

[3] See http://www.hl7.org/implement/standards/product_brief.cfm?product_id=280.

types direct the overall access to information in hospitals. Conflicts between policy types are expressed as rules.

Future work will include policy conflict handling that is based on the defined ontologies. Moreover, prototypical tool support for policy authoring is intended. Ideally, (valid) policies are stored in a repository which serves as a single point of access when new organizational requirements should be translated into guidelines and rules in IT management.

References

[1] L. Aversano, C. Grasso, and M. Tortorella, "A Literature Review of Business/IT Alignment Strategies," *Procedia Technology*, vol. 5, pp. 462–474, 2012.

[2] J. Caumanns, R. Kuhlisch, O. Pfaff, and O. Rode, "IHE IT-Infrastructure White Paper: Access Control," IHE International, Sep. 2009.

[3] R. Wies, "Policy Definition and Classification: Aspects, Criteria, and Examples," in *Proceedings of the IFIP/IEEE International Workshop on Distributed Systems: Operations & Management,*, Toulouse, France, 1994, pp. 10–12.

[4] D. F. Sterne, "On the buzzword ``security policy'','' in *Proc. 1991 IEEE Symposium on Research in Security and Privacy*, Oakland CA, 1991, pp. 219–230.

[5] J. R. Vacca, *Computer and Information Security Handbook*. Morgan Kaufmann, 2009.

[6] International Organization for Standardization, "Health informatics – Privilege management and access control – Part 2: Formal models," ISO/TS 22600-2:2006(E), Aug. 2006.

[7] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser, "Terminology for Policy-Based Management," Internet Engineering Task Force, RFC 3198, Nov. 2001.

[8] J. D. Moffett and M. S. Sloman, "Policy Conflict Analysis in Distributed System Management," *Journal of Organizational Computing*, vol. 4, no. 1, pp. 1–22, 1994.

[9] D. Daiqin He and J. Yang, "Authorization Control in Collaborative Healthcare Systems," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 4, no. 2, pp. 88–109, Aug. 2009.

[10] J. Krogstie, *Model-Based Development and Evolution of Information Systems - A Quality Approach*. Springer London, 2012.

[11] B. Thalheim, "The Concepton of the Model," in *Business Information Systems: 16th International Conference, BIS 2013, Poznań, Poland, June 2013, Proceedings*, W. Abramowicz, Ed. Springer, 2013, pp. 113–124.

[12] W. Yung, "Bring existing data to the Semantic Web: Expose LDAP directories to the Semantic Web with SquirrelRDF," *IBM developerWorks*, 01-May-2007. [Online]. Available: http://www.ibm.com/developerworks/xml/library/x-semweb/index.html. [Accessed: 06-Jul-2013].

[13] Conference of the Data Protection Commissioners of the Federation and the Federal Länder, "Orientierungshilfe Krankenhausinformationssysteme," in *Datenschutzkonforme Gestaltung und Nutzung von Krankenhausinformationssystemen*, Würzburg, 2011.

[14] T. Moses, “eXtensible Access Control Markup Language (XACML),” OASIS, oasis-access_control-xacml-2.0-core-spec-os, Feb. 2005.

[15] B. Kempter and V. Danciu, “Generic Policy Conflict Handling Using a priori Models,” in *Ambient Networks, 16th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2005, Barcelona, Spain, October 24-26, 2005, Proceedings*, vol. 3775, J. Schönwälder and J. Serrat, Eds. Springer Berlin Heidelberg, 2005, pp. 84–96.

[16] M. S. Aphale, T. J. Norman, and M. Şensoy, “Goal-Directed Policy Conflict Detection and Prioritisation,” in *Coordination, Organizations, Institutions, and Norms in Agent Systems VIII*, H. Aldewereld and J. S. Sichman, Eds. Springer Berlin Heidelberg, 2013, pp. 87–104.

[17] M. M. Casalino, H. Plate, and S. Trabelsi, “Transversal Policy Conflict Detection,” in *Engineering Secure Software and Systems*, G. Barthe, B. Livshits, and R. Scandariato, Eds. Springer Berlin Heidelberg, 2012, pp. 30–37.