Interoperable Semantic Access Control for Highly Dynamic Coalitions

Hristo Koshutanski and Antonio Maña
{hristo,amg}@lcc.uma.es

Computer Science Department
University of Malaga (Spain)

Abstract. A coalition consists of independent organizations that share resources and skills to achieve significant mission objectives. Dynamic coalition formations occur in response to some market demands, business requests, or disaster responses, to name a few. Partners forming a coalition are automatically selected given some business criteria and become active participants from the time the coalition is formed. Highly dynamic coalitions (HDCs) form a sub class of dynamic coalitions where the coalition formation and operation are strictly bound by time in order to provide a prompt reaction to some events. This type of dynamism poses the necessity of underlying security models and technologies allowing for automated coalition formation and operation.
This paper presents a platform-driven approach to HDCs. It first defines a life cycle inherent to HDC formations, and then presents a platform-driven access control model that takes advantage of semantics of partners' requirements to provide interoperable access control to resources shared in a coalition. Coalition partners can achieve a high level of service interoperation by enhancing their access control requirements with semantics of usage, and interlinking their semantics using class relations based on standard ontology.

1 Introduction

A Dynamic Coalition (DC) as defined in [1] is "the means through which a group of entities with common interest collaborate to achieve significant mission objectives". This type of coalitions allow small and medium enterprises to be more innovative and competitive in the market[1] but also open the possibility of providing combined services adapted to some client's requests. The dynamism in this case is often limited, as normally the coalition is static after it is formed, but sometimes the dynamism is extended to cover partner replacement during coalition operation. However, time scales for coalition formation and restructuring are long, thus allowing that these processes include some human intervention or complex adaptation process. A DC shares the view of open systems where resources are available to a wide range of clients or often to potentially unknown clients.

In Dynamic Virtual Organizations (DVOs) [2, 3] membership and structure may evolve over time to accommodate changes in requirements or to adapt to new opportunities in the business environment. Therefore, the dynamism in this case is considered throughout the whole life-cycle of the coalition, but still no strong timing constraints in the formation or restructuring of the DVOs are normally required.

In this paper we target a subtype of DCs and DVOs that we will call Highly Dynamic Coalitions (HDCs). The main characteristic of HDCs is their short life, which introduces the need for on-the-fly formation and restructuring of the coalition by means of computer and communication systems in order to respond to clients' requests under strong timing constraints. This type of dynamic coalitions provide important advantages in different scenarios, ranging from continuous service provisioning and business-oriented DCs to incident and disaster response[2].

A coalition instance is formed dynamically triggered by a coalition formation request. Partners are automatically selected given some business criteria and become active participants from the time the

[1] Digital Ecosystems EU initiative – http://www.digital-ecosystems.org; EU FP6 ECOLEAD project – http://ecolead.vtt.fi
[2] EU FP6 OASIS project – http://www.oasis-fp6.org

coalition is formed. Depending on the coalition business model, after a coalition formation took place, the coalition instance is either: (i) automatically executed as an end-to-end business process execution, or (ii) executed as a single-service business process by an end-user request, or (iii) being executed as individual services by end-users on demand. The first case gives the most prompt business reaction with only partner-based interactions. The second case extends the first case but with an end-user triggered process execution. While the third case defines a coalition formation as a set of services being provided to end-users. Whether the set of services are individual partners' services or some services are provided as a composition of partners' individual services is defined by the business model behind a coalition.

We based our model on the existence of a coalition platform that provides means to support coalition formations and operations, including coalition-centric access control enforcement of partners' requirements. However, it is important to note that most likely partners participating in a coalition will have heterogeneous access control models in terms of syntactic and schematic description. This makes service interoperation hard and, sometimes, practically impossible to achieve.

The adoption of a platform-centric model for coalition formations and operations allows partners to achieve better coherence of their service usage during coalition operation. The contribution of this work is twofold:

- Identifying the inherent life cycle of HDCs along with research challenges to each phase of the life cycle,
- Defining a platform-driven access control model based on semantics of partners' access control specifications to provide interoperable access control process to resources shared in a coalition.

The life cycle gives us understanding of the foundations and potential of HDCs, and, at the same time, positions interoperable access control throughout the life cycle. The paper main contribution is on defining a model that unifies the semantics of partners' access control requirements and provides a coherent evaluation and enforcement of those. Clients can use coalition resources with minimal burden of satisfying partners' requirements as the platform can take an informed and automated decision on whether to provide access to the service or not based on these semantics. An automated credential negotiation process between an end-user and a coalition instance is described that takes advantage of coalition-specific semantic interoperability of credentials.

An interesting aspect as a result of the access control interoperation is the provision of state-based semantic interoperability of requirements, for example, normal state coalition operation versus critical or emergency states, or open business operation versus VIP-based operation, etc. In such state cases entities recognized with certain credentials are provided access to more or less services depending on the defined state-based interoperability, but without the necessity of restructuring individual partners' access control policies.

Summary of model contribution. The proposed access control model targets the problems of (i) distributing the responsibility of access control aspects between the coalition platform and partners; (ii) reconciling different access control policies of partners; (iii) ensuring semantic interoperability of authorization models used by partners; and (iv) allowing establishment of access control mechanisms to coalition resources without costly reengineering of partners' systems and without time-consuming adaptation processes that would be unacceptable during coalition operation.

Section 2 defines the envisaged life cycle of HDCs, and identifies the access control challenges on each phase. Section 3 presents the semantic access control model, how coalition-level interoperation is faced, and evaluation assessment of a coalition access decision process. Section 4 describes an approach on automated credential establishment with coalition end-users, and its extension to semantic interoperability. Section 5 shows the envisaged access control architecture and its functional components. Section 6 discusses related work on semantic access control approaches, and outlines existing technology standards for coalition business modelling. Section 7 concludes the paper.

There are two appendixes included for convenience of readers. Appendix A shows an example formalization of a coalition access control process. Appendix B describes a possible instantiation process of a coalition-wide semantic policy vocabulary based on an RBAC specification.

2 Highly Dynamic Coalitions Life Cycle

To introduce the access control model and its specifics, we will first describe a life cycle of HDCs. This will give us better understanding of the foundations and scope of HDCs. We will identify the main access control challenges inherent to the life cycle, while in subsequent sections we will explore the access control model in details. Along with the access control challenges, we will list some other challenges we found important to better define the foundations of HDCs. Since DCs and VOs cover a broad domain of research, there are several important aspects not discussed in this paper, such as legal[3] and social aspects (e.g., [4]), trust and security aspects (e.g., [2, 5–7]), and reputation aspects (e.g., [8]).

Figure 1 shows the main phases of HDC life-cycle. A coalition platform provides a means to support coalition formation, operation and closedown. The existence of a platform is essential to the life cycle, as it will comprise all technological aspects described in this section.

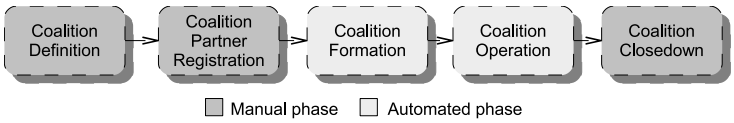


Fig. 1. Highly dynamic coalitions life cycle

2.1 Coalition definition

This first phase deals with the task of defining the coalition model. Before a coalition can be dynamically formed to respond to clients' requests, the coalition platform must have information about the services, workflow and requirements for partners that will participate in the coalition. Therefore, we introduce the concept of a coalition model to represent the set of definitions that describe a type of coalition. Of course, it is possible for a coalition platform to support several coalition models. The different aspects that are part of the coalition model must be defined. For some of them existing approaches provide a suitable solution, although extensions are required, while for other aspects new solutions have to be developed.

A crucial process of the coalition definition phase is the definition of a coalition access policy. The definition of an access control policy for a dynamic coalition is a non-trivial aspect. In fact, this aspect requires a very flexible model for the specification of policies, because these policies must reflect the access control requirements of the coalition, but also those of the partners. Therefore, policy composition and in particular, conflict-free composition is an essential feature of the policy language. It is not feasible in this phase to include the definition of the policies of partners because they are unknown at this stage. Therefore, it is required that the policy specification language has modular capabilities that allow for dynamic and automated composition of the coalition policy during the coalition formation phase. This composition will use the policies of partners that must be provided during partner registration.

Research challenges:

– Defining business models, including requirements for partners, along with their roles and services. We propose adopting an existing solution. However, limitations in the existing proposals for business modeling require the definition of extensions. In particular, requirements defined for partners must be dynamically and automatically verifiable. We need to trust the models of the partners, which in turn calls for certified partners' descriptions. Extensions to business modelling languages are required in order to certify the partners' models. Section 6 overviews existing business modelling languages and their relation with the identified challenges.

[3] http://www.legal-ist.org

- Defining a global access control policy. The main issue in this case is that the policy must take into account the partner's policies, but these are not known at this stage. The key here is to define modular and composable policies. Section 6 discusses [9, 10] as a potential approach for expressing coalition access control policies.
- Abstracting partners' requirements. Opposed to static coalitions where business models can include references to the partners' services that are used, in a dynamic coalition scenario, the specific partners providing such services are only known when a coalition formation request is received. Moreover, it is possible that the set of possible partners is not known at this stage of coalition definition. In consequence, the business model needs to be complemented with models for each of the partners that will participate in the coalition. These partners' models describe requirements for each of the roles that are defined in the coalition model. Modelling of partners, especially taking into account points of view such as trust, security and other qualitative aspects has not received enough attention in the current modelling language support. Additionally, the specific characteristics of dynamic coalitions require the partner model to be flexible in order to contemplate different elements that can be used in the process of selecting partners in the coalition formation phase.

2.2 Coalition partner registration

Before a partner becomes part of a HDC, it must indicate its interest in doing so, i.e., provide some initial information that will be used by the coalition platform in the partner selection process during the coalition formation phase, as well as, information that will be used during the operation of the coalition. Among this information partners must define the services offered, the conditions under which these services are offered, constraints, etc. This is achieved by registering to a coalition platform. Each organization selects which business models it is willing to participate and what business roles to play. This phase is therefore a prerequisite to the use of the coalition platform and may involve non-automatic processes or actions.

In this phase, partners need to first define the target coalitions that they wish to join. This includes selecting the coalition models that the partner wants to participate in. Each partner can join different coalition models, and can do so playing different roles, depending on the services that the partner can and wishes to provide, so this phase also includes the selection of the models and the roles. Moreover, in general, no partner will want to join a coalition unconditionally. Therefore, partners need to express the restrictions that they want to impose in order to join specific coalitions. Furthermore, the coalition platform may also want to impose conditions on the partners in order to allow them to register for a coalition model. For this, the partner has to prove the fulfilment of these conditions. Sometimes, the partner has to accept the rules established by the coalition platform. Finally, partners need to define their access control policies or to make them available to the coalition platform.

We note that a partner can register to coalition models at any time during the platform lifetime, but a partner participation to a given coalition model is either during a coalition formation phase or as a partner replacement during an existing coalition instance operation. There could be more than one existing coalition instances of a same coalition business model if triggered by different requests for formations.

Regarding security settings, it is important to highlight (i) the establishment of access control mechanisms to the organization's resources, and (ii) the translation and adaptation of organization's access policies to the platform's supported access control model(s). When a partner registers to the coalition platform it provides a semantic access policy and a set of semantic relations the partner has agreed on with the other coalition partners. The sets of partners' semantic relations are used by a transformation process to internally generate a coalition semantic interoperability policy.

Another important aspect to consider is the partner registration model, which reflects directly on the degree of dynamism one can achieve during the coalition operation phase. Especially, the synergy of access control information provided by partners during registration will underlie the possible degree of partner replacement during coalition operation.

There are two main approaches for registration of access control models:

- *static approach* – partners mutually agree on specific semantic relations between their access control specifications and credential requirements. In this case, a partner replacement during operation is to be done statically during the coalition formation phase, as partners' access control interoperation depends directly on the other partners' specifications.
- *dynamic approach* – given predefined semantics of credentials (and implications among them) generic for all business models supported by a platform, partners define their specific access control requirements semantically characterized according to the platform's wide semantics of credentials. In this way, a partner can be replaced dynamically since its access control specification does not depend on other partners' specifications in a current coalition formation but on the general semantics provided by the coalition platform. The advantage of this approach is that for a large number of registrants the semantic interoperation of access control is faced on the platform level, via the predefined semantics, so that a partner needs only to reference its access control specification (credential attributes) to the given platform's semantics (avoiding cumbersome bilateral agreements). However, the disadvantage is that dependencies between actual partners' semantic access control requirements are not expressive enough and, as a consequence, seamless cross-partner service usage is not well addressed.

Research challenges:

- Define services offered by a partner. This relates to the coalition modeling mentioned in the previous phase. The focus here is on modeling the services offered by partners in a way that allows the coalition platform to match them to the ones defined for the coalition model.
- Prove fulfilment of requirements. In general this deals with statically verified aspects. Certified partner's descriptions can be used to securely attest these aspects. However, some dynamic aspects (e.g., partner workload) can also be relevant during the coalition formation phase. The certification of these is much more complex and other approaches such as monitoring can be used in this case.
- Define partner's access control policy. This strongly relates to the definition of the global coalition policy described in the previous phase. In this case, we do not start from scratch and, therefore, we can expect that most of the partners have already some access control mechanism in place. The challenge, in this case, is defining automated means to transform the existing access control specification of a partner into a format that the coalition can use, as we shall see in Section 3.
- Define a hybrid approach of partners' access control registration where partners achieve a good semantic synergy between their access control specifications per a coalition model, and at the same time, are easy replaceable with other partners during coalition operation.
- Define target coalitions in terms of partners' restrictions and preferences. During registration, a partner is required to provide information to (i) express the target coalitions in which a partner wants to participate; and (ii) define the restrictions a partner wants to impose on the other partners forming a coalition with it. As in the case of the definition of the requirements for partners, it is also necessary to certify the partners' profiles. Therefore, this information is recorded using secure profiles [11]. In the coalition formation phase, the process of selecting the set of partners chosen by the coalition platform to respond to a request must also take into account those preferences and restrictions. For instance, partner A may want to play role X in coalitions that are used to respond to requests coming from a specific set of clients, but only if role Y in those coalitions is not played by partner B.

2.3 Coalition formation

Once organizations have successfully registered to the platform, they are able to participate in coalition formations. The important characteristic of this phase is the *automated* formation of coalitions for prompt reactions to market requests based on the means established in the previous phase. A research direction is to characterize what a suitable model is (e.g., [12, 13]) and how to adapt it to the coalition platform foundations. We believe that BPEL-based business models can provide a sound foundation for this process. Section 6 overviews some of the existing business modelling languages. In any case, the process

must take into account the different elements of a coalition business model, partners' models, partners' requirements and preferences, etc.

Research challenges:

- Define what information to be provided in coalition formation requests. The request must contain information to help determining the best set of partners to be part of the coalition and is therefore essential for the process of coalition formation. Additionally, different means of activating coalition formation requests should be provided, e.g. by defining appropriate interfaces for that, for example, as Web services interfaces and invocation mechanisms. Important issue here is the possibility of remote and mobile activation of coalition formation requests.
- Translation and integration of partners' access control specification. This aspect is related to the definition of the partners' policies described in the previous phase. However, in this case the focus is on the dynamic creation of a specific coalition policy based on (i) the global coalition policy, and (ii) the selected partners' policies and context conditions.
  A specific set of access control vocabulary per coalition platform will allow partners to "transform/translate" their access control specifications to unambiguous semantic description, which in turn is uniformly enforced by the platform's security manager.

2.4 Coalition operation

This phase deals with the fulfillment of HDC goals and the provision of the desired services. To achieve these, the coalition platform must provide operations of the individual organizations participating in the DC by complying with their security policies. Although it may intuitively appear that this phase is the most complex from access control point of view, the DC operation can be quite simple if the previous two phases have been adequately carried out.

Research challenges:

- Policy evaluation and enforcement. This can be executed either at the coalition platform, or at partners' side. It can even be distributed among them. There are reasons to assign it to both sides, so we need to find a compromise. Our approach here is to define this aspect as part of the access policies expressed both at coalition level and at partners' levels. Section 6 describes a model and a language [9, 10] well suited to support the expression of coalition policies, allowing us to define extensions to control where policies (or part of them) are located.
- Degree of dynamism of coalitions. Dynamic coalition operation in terms of automated, on the fly, partner selection and replacement is a challenging part on its own. Regarding our model, we are interested in the dynamic and automated partners replacement, and how to address it from an access control point of view. The solution to this challenge depends on the access control model (i.e., what access control specification a partner supplies to a coalition platform), and the use of semantics of partner's access control requirements and how these semantics interoperate with those of the other partners in a current coalition formation.

2.5 Coalition closedown

The objective of this last phase is to finalize the operation of a HDC. It may entail recording information about the life of the coalition, distributing the benefits, etc. In this phase the envisaged security aspects are focused on the protection of the information recorded by the DC.

Research challenges:

- Secure recording of coalition history. In some settings, this may be required for operational, strategic, legal or financial purposes.

– Distribution of revenues among partners when appropriate.
– Coalition models evolution and optimization. The information obtained during the operation of a specific coalition can be used for the selection and operation of subsequent coalition instances.

3 Interoperable Semantic Access Control

This section presents the foundations of the semantic access control model for HDC. It will first intuitively define the core element of the semantic access control model, that is, the semantics of credential usage in access policies. It will summarize what a semantic access policy is with respect to a partner's intention behind, and then will provide the core notion of a coalition-wide semantic policy vocabulary for the partners' semantic access policies. We assume that each partner in a coalition has requirements in terms of credentials. A prerequisite of PKI/PMI [14] is essential for (remote) credential verification and policy enforcement by a coalition platform.

The model discussed in this section corresponds to the challenges identified in the partner registration phase, where each partner defines its semantic access policy, and a corresponding policy instantiation using the coalition-wide policy vocabulary. The model is also relevant to the coalition formation and coalition operation phases. This section will also present the foundations of coalition-level semantic interoperability of credentials, as an inherent step to an interoperable semantic access control process, part of the coalition operation phase. Partners establish a coalition layer credential interoperability by bilateral/multilateral agreements, and as a second step of partners' access control registration process, i.e., a step after the partners' definition and instantiation of their semantic access policies.

3.1 Semantics of credential usage in access policies: intuition

A credential encapsulates the notion of a digitally signed document attesting that a holder entity has an (identifiable) attribute, and that the credential statement is issued (digitally signed) by a certificate authority (a provider). A credential attests that a holder has certain attributes, which is enough in scenarios where the semantics of each attribute is well-defined.

However, in a multy-party scenario, such as HDCs, where different partners with heterogeneous access and authorization models interact with each other, a credential (i.e. its attribute value) may have different semantics when used in different contexts. On the other hand, different credentials issued by different parties may have semantically equivalent values in certain contexts. Therefore, an important step toward access control interoperability is the definition of semantics of credentials in access policies. We define a semantic access policy to be an access policy where each credential is used along with a given semantic context. The policy structure does not change, but a semantic term is coupled with a credential. Let us illustrate it with the following example.

Example 1. Partner A defines that service1 is granted if an e-drivingLicense is given. Partner A's intention for that is to attest that the client is a legal entity in the context of being over 18. A semantic access policy would look like: service1 is granted if a tuple ⟨e-drivingLicense, over18⟩ is given.

Despite of its simplicity, the example gives us the intuition of how a semantic access policy would be derived from a given (syntactic) access policy and a set of semantic contexts. Yet, one could define that a credential keeps its inherent semantics, such as service1 is granted if an ⟨e-drivingLicense, e-drivingLicense⟩ tuple is given. Therefore, the meaning of the tuple in an access control policy is to explicitly define the semantics (the policy designer's intention) of the usage of a given credential.

Example 2. Let Partner B be a university organization that provides some services to graduate students. Partner B defines that service2 is granted if a credential for a bachelor degree is given. Partner B's intention for that is to attest that the user is a graduate student. Partner B's semantic access policy would look like: service2 is granted if ⟨bachelorDegree, graduateStudent⟩ tuple is given.

Considering examples 1 and 2, the two contexts for over18 and graduateStudent partially overlap their semantics but use different credentials to prove those semantics. The intuition here is to define an interoperability notion of credential equivalence for decision making based on the semantics of credentials usage in access control policies. Section 3.3 formally defines the semantic interoperability issue.

Now, if partners A and B are to join a coalition, and given their semantic policies, partner A can agree with B to link the two semantic contexts with the following relation *B.graduateStudent subClassOf A.over18*. From that on, credentials being assigned semantic context graduateStudent by B have semantically equivalent (access control) value with credentials being assigned semantic context over18 by partner A. In this way, partner A increases the potential set of clients to its services by exploring the possible relations of credential semantics within a coalition.

Each partner defines the exact semantics of credentials used in its access control policy. At organization registration phase all partners in a coalition mutually agree on possible semantic relations with each other. Any relation is based on bilateral agreement between partners, which decide whether to limit the credential equivalence within the bilateral relation only or to allow credential equivalence by transitive relations with other partners' semantic contexts.

Since a credential could be assigned different semantic contexts intrinsic to the access policy it is used into, for example, e-drivingLicense to identification, citizenship, over 18 or over 12 in case of movie restrictions, driving skills etc., coalition partners could either mutually agree when their credential semantics interoperate with each other (with most accuracy), or, instead, individually define how their credential semantics interoperate with the platform's predefined semantics (with less accuracy of interoperation between partners).

The model also facilitates the usage of semantic contexts when expressing access control constraints so that one can define policy constraints on particular semantics of credential usage. However, care must be taken when dealing with semantics for policy constraints as they express restrictions over the entire access policy, i.e. over the semantics of all credentials in the policy. The issue of how to enforce policy consistency depends on the security model behind an access policy specification.

By using semantics of credentials the model enforces local policy consistency against coalition-wide credential interoperability. Thus, a client having credentials from organization A but requesting a service at organization B may be recognized in B with privileges that are in conflict with B's local policy.

Given an ontology, we define semantics for an access policy as a set of terms that give meaning to credentials used in the policy. Semantics of an access policy may differ from a coalition to coalition but, in all cases, the semantic access policy preserves the original structure of the (syntactic) access policy it is derived from (if not directly defined).

Definition 1. (Semantic Access Policy) *Let $\mathcal{P}_{\mathcal{A}}$ be an access policy of a partner sharing resources in a coalition and let $\mathcal{O}$ be a set of contexts giving semantics to credentials used in $\mathcal{P}_{\mathcal{A}}$. We say that $\mathcal{P}_{\mathcal{SA}}$ is a semantic access policy of $\mathcal{P}_{\mathcal{A}}$ with respect to $\mathcal{O}$ if any credential term occurring in $\mathcal{P}_{\mathcal{A}}$ is assigned a semantic context of $\mathcal{O}$ in $\mathcal{P}_{\mathcal{SA}}$.*

We note that the definition provides a generic notion of semantic access policy since the semantic assignment is performed on any credential usage in both the grant requirements part and constraints part of a policy. The actual semantic assignment may depend also on a security model behind the policy.

Client's set of credentials have syntactic values that obtain semantic values by assigning contexts to them. Each partner assigns semantics to any credential used in its access policy, forming its, partner-specific, syntactic to semantic credential assignment. A coalition credential semantic assignment is defined by the union of all partners' semantic assignments. A syntactic credential term can result as being assigned multiple contexts within a single partner's policy or to several semantic contexts across multiple partners' policies in a coalition.

3.2 Coalition-wide semantics of partners access policies

A coalition platform is intended to provide coalition formation services to partners from different administrative domains. Because partners have heterogeneous requirements in terms of description and definition,

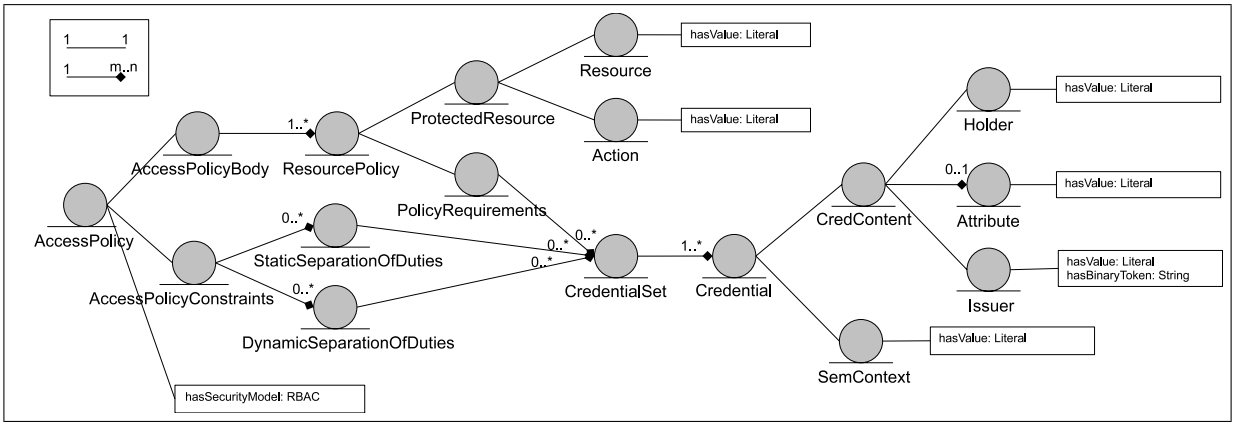


Fig. 2. Example of coalition policy vocabulary for RBAC-compliant models

the platform model provides the notion of *coalition policy vocabulary*. The coalition vocabulary aims at defining a coalition-wide semantics (ontology) of an access policy structure that partners will use to instantiate their own requirements for the resources they share. During the partner registration phase a partner provides its semantic access policy in accordance with the coalition policy vocabulary.

The coalition-wide policy vocabulary has the purpose to unifying different partners' requirements for the sake of unambiguous evaluation by a coalition policy decision point (PDP). It is used as metadata that provides a common vocabulary and a policy structure that partners have to conform to when defining their own policies. The vocabulary is published along with a human language description explaining its exact meaning in order to facilitate a policy instantiation process.

Figure 2 shows a coalition access policy structure and vocabulary of RBAC-compliant models. The policy structure can be expressed in OWL [15] and instantiated by means of standard ontology annotation tools, such as Protégé[4].

The policy structure has as root an AccessPolicy class that has two subclasses – AccessPolicyBody and AccessPolicyConstraints – and a property hasSecurityModel that defines a choice of supported security models at a coalition. The AccessPolicyBody class has one or more ResourcePolicy classes. A ResourcePolicy class encapsulates a ProtectedResource and its PolicyRequirements. The ProtectedResource class encapsulates a Resource identifier and an Action defined on it. Both Resource and Action classes have generic values of type Literal.

The PolicyRequirements class encapsulates all credential requirements for the ProtectedResource. It has zero or more StaticSeparationOfDuties and DynamicSeparationOfDuties classes each of them containing zero or more CredentialSet classes. Each CredentialSet class has one or more Credential classes, where a Credential class encapsulates all credential information necessary for the coalition evaluation process. The Credential class consists of a CredContext class – defining the actual credential content – and a SemContext class – defining the semantic context the credential content is used into. A CredContent is defined by having Holder, Attribute, and Provider classes. Each of the classes has a value of a type Literal. If a CredContent class has no Attribute class defined the Credential class refers to an identity token. The credential Provider class has a property hasBinaryToken that encapsulates the provider's security token trusted to sign the Attribute class. This property is used to supply to a coalition platform the set of trusted certification authorities of a given partner's policy.

The semantics behind the coalition policy structure is the following. Policy requirements for a resource define disjunctive credential sets where a credential set contains all necessary credentials that grant the resource. If no credential set then the resource is unprotected. In contrast, policy constraints define

[4] http://protege.stanford.edu

conjunctive credential sets of static and dynamic separation of duties, where each credential set defines a credential configuration that violate access policy consistency.

The policy vocabulary reflects the supported access control models by a given coalition platform. The just described policy structure best reflects RBAC-compliant policies. However, one can extend the vocabulary to handle non-RBAC notions such as negation as failure on policy requirements, negative authorization statements along with supported policy resolution rules, or policy combination rules in case partners' resources interact with each other.

A semantic access policy could directly be specified at the (remote) coalition platform or bootstrapped from a partner's existing access control specification. Appendix B shows a possible instantiation process of the above described policy vocabulary based on the RBAC model specification.

The underlying semantic policy reasoning. During registration, when a partner has defined (instantiated) its semantic access policy it uploads the policy to a coalition registration agent/service. The policy is then validated against the coalition policy vocabulary and transformed into a suitable representation for subsequent policy analysis and reasoning.

Access policies (and in general trust management) languages need a declarative and formal foundation for reasoning and evaluation [16]. Datalog has shown to be a promising formal foundation for policy specification and reasoning. It has been the foundation of several trust management languages, such as Delegation Logic [17], the RT (Role-based Trust-management) framework [18], SD3 (Secure Dynamically Distributed Datalog) [19], and Cassandra [20]. We adopt Datalog and logic programming [21, 22] as an underlying policy representation and reasoning. Partners' semantic access policies are transformed to logic programs for subsequent evaluation and enforcement. A logic program is a set of rules of the form:

$$A \leftarrow B_1, \ldots, B_n,\ not\, C_1, \ldots,\ not\, C_m$$

A is called the head of the rule, each B_i is called a positive literal and each $not\, C_j$ is a negative literal, whereas the conjunction of B_i and $not\, C_j$ is called the body of the rule. If the body is empty the rule is called a fact.

One of the prominent semantics for normal logic programs is the stable model semantics [23]. The intuition is to interpret the rules of a program P as constraints on a solution set S (a set of ground atoms) for the program itself. So, if S is a set of atoms, the above rule states that if all B_i are in S and none of C_j are in it, then A must be in S. In our model we also need constraints that are rules with an empty head.

$$\leftarrow B_1, \ldots, B_n,\ not\, C_1, \ldots,\ not\, C_m$$

A constraint rules out from the set of acceptable models situations in which all B_i are true and all C_j are false.

Let C be a tuple encapsulating the values a credential token contains, in our case $C = \langle \mathsf{Holder}:h, \mathsf{Attribute}:a, \mathsf{Provider}:p \rangle$. We will use C as an abstraction of a credential token to provide a concise view of credential information used in the logical model.

- $\mathsf{grant}\,(\mathsf{Resource}:res, \mathsf{Action}:act)$ a predicate denoting when an $\mathsf{Action}:act$ is granted to be performed on a $\mathsf{Resource}:res$. Depending on the specifics of shared resources additional information can be included in the resource.
- $\mathsf{cred}\,(C)$ a predicate representing a credential term with values encapsulated by the tuple C.
- $\mathsf{sem\_cred}\,(C, \mathsf{Context}:O)$ a predicate representing a semantic credential term that associates a credential C (identified by its values) to a semantic context O.

To transform a semantic policy to a logic policy we generate a (grant) logic rule for any credential set element in the policy requirements part and a constraint rule for any credential set in the access policy constraint part of the access policy body (rf. Figure 2). Thus, for any CredentialSet element of PolicyRequirements part we generate:

$$\mathsf{grant}\,(res, act) \leftarrow \mathsf{sem\_cred}\,(C_1, O_1), \ldots, \mathsf{sem\_cred}\,(C_n, O_n).$$

where the head of the rule represents the ProtectedResource element and the body all Credential elements part of the CredentialSet. Likewise, for any CredentialSet of StaticSeparationOfDuties and DynamicSeparationOfDuties we generate a constraint rule of the form:

$$\leftarrow \mathsf{sem\_cred}\,(C_1, O_1), \ldots, \mathsf{sem\_cred}\,(C_n, O_n).$$

The constraint rule would syntactically differ with a session information difference when dynamic or static separation of duties are expressed.

Partners' access policies have a security model underlying their semantics. Any supported security model defines specific relations among credentials (their content) in generated rules. To this extend, the previously described policy vocabulary targets RBAC models where we have role sets relations (users to roles and roles to permissions) and a hierarchy of them. Appendix B shows a possible algorithmic process of deriving an instance of the coalition policy vocabulary from an RBAC specification.

However, the proposed coalition vocabulary structure is unable to capture generic attribute-based access control models such as RT_0 [24]. To approach general attribute-based access control specifications, one needs to either extend the coalition vocabulary to capture possible rule-based policy specifications (for example, allowing the expression of the four types of rules in RT_0) or adopt a semantic Web policy language [25] allowing for the expression of different access control models.

A recent work [26] provides a comprehensive treatment of representing RBAC models [27, 28] using the Web Ontology Language [15]. There are two main advantages of using OWL for policy definition, as authors define. First advantage is facilitating the semantics of what policy objects are, especially when shared across multiple partners (e.g., what refers to a "public printer" object or a "full time student" role, etc). Second advantage is that OWL is grounded in Description Logic (DL) [29] that facilitates the translation of policies expressed in OWL to other formalisms for subsequent analysis or execution. The authors also give discussion beyond RBAC models, where the classical OWL DL reasoners cannot capture some trust management concepts.

A research challenge is to provide coalition-wide OWL-based descriptions of access control models with an appropriate translation to datalog with constraints as a foundation of policy analysis and reasoning [16]. A suitable for our coalition approach is the OWL Flight language [30]. OWL Flight is a variant of OWL but based on Logic Programming rather than DL. Although, with OWL Flight constructs we loose some of the expressive power of OWL DL (because of disjunction and existential quantification allowed in OWL DL), but we benefit of having efficient query answering which is an important aspect of an access decision process. OWL Flight provides close expressiveness to that of OWL DL in context of possible access control requirements specifications, but with well-defined translation to datalog programs with integrity constraints and default negation.

3.3 Semantic interoperability of credentials

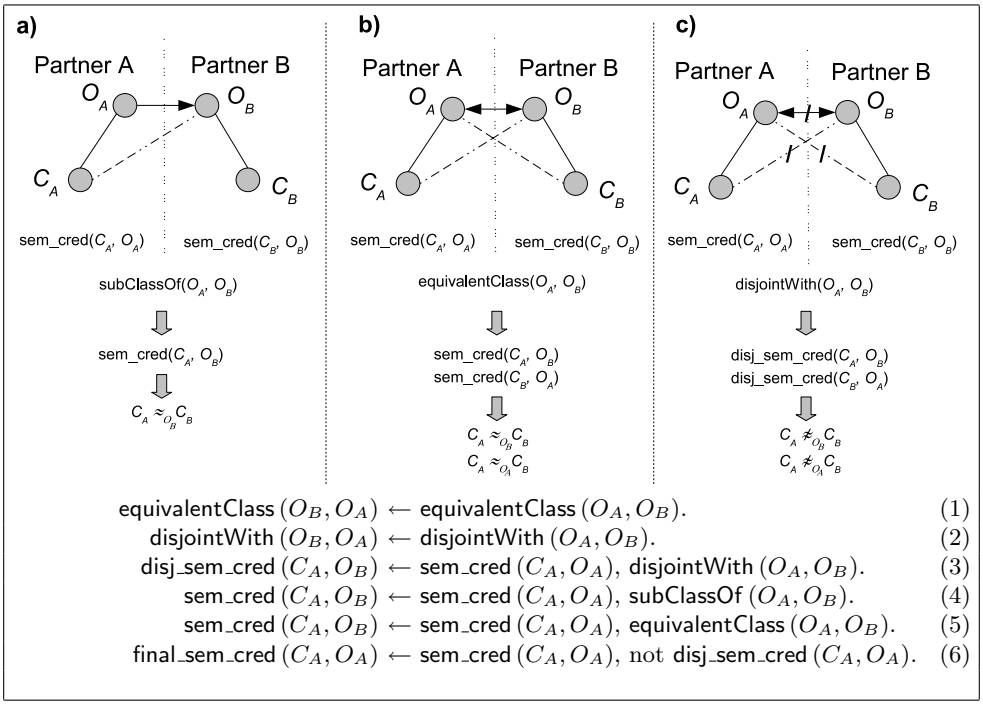
Along with the semantic policy, each partner provides a set of semantic implications of credentials it has agreed on with other coalition partners. The sets of semantic credential relations are (internally) transformed to a coalition semantic credential interoperability policy. This section looks in details of the foundations of semantic credential implications and how these credential implications provide access control interoperation during the coalition operation phase.

A semantic credential interoperability defines credential implications between semantically related contexts. Partners define semantic contexts of their (local) access policies in accordance with their (local) ontology. At registration phase coalition partners relate each other's contexts based on the semantics behind them.

The main intuition when defining an interoperability notion is that assigning a credential to a context is interpreted as the credential is a member of that context and, inspired by the semantic Web paradigm, the credential can be seen as an instance of that context when used in access policy definitions. Therefore, the semantics of credential usage in a policy is the semantics of the context(s) the credential belongs to. The coalition credential interoperability notion is to allow credential entailment based on relations between partners' semantic contexts to which credentials are members of.

Definition 2. (Context-based Semantic Credential Equivalence) *Let o be a given semantic context and let c_i and c_j be two credential terms. We say that c_i and c_j are semantically equivalent in context o, $c_i \approx_o c_j$, if they are both assigned members of o. If $c_i \approx_o c_j$ then it also holds that $c_j \approx_o c_i$.*

The notion of semantic credential equivalence defines when a credential is equivalent to another one in the semantics of a given context. We note that two credentials could be equivalent in one context and disjoint in another context. When two credentials are equivalent in a context then either of the credentials implies the other one under that context.



$$\mathsf{equivalentClass}\,(O_B, O_A) \leftarrow \mathsf{equivalentClass}\,(O_A, O_B). \quad (1)$$
$$\mathsf{disjointWith}\,(O_B, O_A) \leftarrow \mathsf{disjointWith}\,(O_A, O_B). \quad (2)$$
$$\mathsf{disj\_sem\_cred}\,(C_A, O_B) \leftarrow \mathsf{sem\_cred}\,(C_A, O_A), \mathsf{disjointWith}\,(O_A, O_B). \quad (3)$$
$$\mathsf{sem\_cred}\,(C_A, O_B) \leftarrow \mathsf{sem\_cred}\,(C_A, O_A), \mathsf{subClassOf}\,(O_A, O_B). \quad (4)$$
$$\mathsf{sem\_cred}\,(C_A, O_B) \leftarrow \mathsf{sem\_cred}\,(C_A, O_A), \mathsf{equivalentClass}\,(O_A, O_B). \quad (5)$$
$$\mathsf{final\_sem\_cred}\,(C_A, O_A) \leftarrow \mathsf{sem\_cred}\,(C_A, O_A), \text{not } \mathsf{disj\_sem\_cred}\,(C_A, O_A). \quad (6)$$

Fig. 3. Coalition context relations and semantic credential implications

Definition 3. (Coalition Semantic Context Relations) *Let $\mathcal{X}$ be a coalition instance defined by a set of partners and let $\mathcal{O}^i = \{O_1^i, \ldots, O_n^i\}$ be the set of semantic (ontology) contexts a partner i defines ($i \in \mathcal{X}$). We denote $\mathcal{R}$ to be the set of all bilateral context relations defined by partners in a coalition where any relation has a type one of the following class relations [15]:*

- $\mathsf{subClassOf}\left(O_i^a, O_j^b\right)$ *denotes when context (class) O_i^a is a subclass of context O_j^b, i.e. elements of O_i^a are also elements of O_j^b ($O_i^a \subseteq O_j^b$). $a, b \in \mathcal{X}$ indicates partner a' and partner b's local settings, where a and b might be a same partner.*
- $\mathsf{equivalentClass}\left(O_i^a, O_j^b\right)$ *denotes when context O_i^a have the same element instances with context O_j^b, i.e. ($O_i^a \equiv O_j^b$). a and b might be a same partner.*
- $\mathsf{disjointWith}\left(O_i^a, O_j^b\right)$ *defines that contexts O_i^a and O_j^b have no elements in common, i.e. ($O_i^a \cap O_j^b = \emptyset$). a and b might be a same partner.*

Any two semantic contexts are related with each other if either of them is an instance of the same type SemContext of the coalition policy vocabulary. We assume that any class relation between two contexts in a coalition is agreed between the two partners holding the semantic contexts and that each partner

relates only its local contexts to other partners' contexts. In this way, we avoid initial inconsistency in coalition context relations.

DisjointWith relation provides the notion of separation of concepts where partners explicitly specify that such a separation is to be enforced. A similar concept is the separation of duty in RBAC. However, in contrast to RBAC, the enforcement of disjointWith relation will take precedence over the assignment of semantic credentials (done by relations subClassOf or equivalentClass) and will not be treated as making the policy inconsistent. The purpose of credential interoperability is to define all possible credential implications among coalition relations.

Figure 3 shows the class relations between partners' contexts and their respective credential implications. O_A and O_B denote semantic context instances of partner A and B, respectively. C_A and C_B denote credential concepts assigned to the contexts O_A and O_B, respectively. Case (a) shows when semantic context O_A is defined as subClassOf relation with context O_B. Then any credential assigned to O_A, i.e. being a member of O_A, becomes also a member of O_B. In this case, credential C_A takes semantics O_B when referred to partner B's local access control settings (ontology). Following Definition 2, credentials C_A and C_B become with equivalent value under the semantics of context O_B.

Case (b) is an extension of case (a) when O_A and O_B have an equivalentClass relation. In this case, credentials C_A and C_B become with equivalent values under the semantics of contexts O_B and O_A. EquivalentWith relation defines symmetric credential interoperability between contexts.

Case (c) shows when O_A and O_B have disjointWith relation. We introduce a new credential term $\mathsf{disj\_sem\_cred}\,(C, O)$ denoting when a credential concept C must not be assigned to a semantic context O. DisjointWith defines symmetric disjoint relation between contexts.

Figure 3 shows also the logic rules that define the behavior of the just described credential implication cases. Rules (1) and (2) define the symmetry property of equivalentClass and disjointWith relations. Rule (3) formalizes the behavior of case (c). It states that if a credential C is assigned to a semantic context O which has a disjointWith relation with O' then a predicate denoting that C is disjointWith O' is entailed. Rule (4) defines the behavior of case (a). It states that if a credential C is assigned to a context O which has a subClassOf relation with a context O' then the rule entails a predicate denoting C assigned to context O'. Similarly, rule (5) holds for case (b). Rule (6) enforces the behavior of disjointWith relations. It states that a final entailment of semantic credentials out of all interoperability relations is enforced on those terms that have not been marked as disjoint with. We use the term $\mathsf{final\_sem\_cred}\,(C, O)$ to denote those.

The just described logic rules are to be complemented by each partner's (local) semantic credential assignments defined along with the partner's semantic access policy. Then, using the coalition semantic context relations, the above rules will derive all final semantic credential assignments supporting any access decision process to coalition services.

We note that although rule (6) uses negation-as-failure in its body, a resulting logic program of an access control process will be a stratified logic program, which ensures a unique stable model for the decision making. Given that partners' semantic credential assignments and coalition semantic relations are facts in a logic program then a resulting logic program complemented with rules (1)–(6) will have no recursion through negation (refer to [22, page 556]) and preserve its stratification.

Key issue for adoption of the model is the possibility of relating partners' contexts at a coalition registration phase. One would argue if two contexts can be unambiguously linked with one of the above class relations. For example, two semantic context might be neither equivalent nor defined as sub class of one of the other. In this case either of the partners can define a new (artificial) context, then they agree on their semantics, and relate them as intended. The new contexts are then coupled with respective (local) credentials according to their new semantics.

Another possibility is to specify restrictions on a coalition relation that define what credentials are to be assigned to either of the contexts via that relation. Such requirements would filter out those credentials that do not hold. For example, if a credential provider is not among a given list of accepted credential providers the assignment of that credential to an equivalent semantic context does not hold.

State-based semantic interoperability. A coalition may have explicit states that characterize a particular functionality. An example could be a critical and non-critical states, or emergency and normal functioning, etc. One can assign state identifiers to coalition relations that hold for a given state. State information could be provided as a third arity in the above class relations, such as equivalentClass(O_i^a,O_j^b,$StateID$), subClassOf(O_i^a,O_j^b,$StateID$), and disjointWith(O_i^a,O_j^b,$StateID$).

Having state-based semantic interoperability one can achieve scalable and automated coalition access management, as credentials will have access to more or less services depending on the coalition operation state. For example, equivalentClass($FirebrigadeOfficer$,$StatepoliceOfficer$,$emergency$) would activate an equivalent relation of a semantic context of a "FirebrigadeOfficer" with that of a "StatepoliceOfficer" in case of an emergency state is triggered. A possible modification of the rules (1)–(6) to handle state-based semantic implications is the following:

$$\mathsf{equivalentClass}\,(O_B, O_A, St_i) \leftarrow \mathsf{equivalentClass}\,(O_A, O_B, St_i).$$
$$\mathsf{disjointWith}\,(O_B, O_A, St_i) \leftarrow \mathsf{disjointWith}\,(O_A, O_B, St_i).$$
$$\mathsf{disj\_sem\_cred}\,(C_A, O_B) \leftarrow \mathsf{sem\_cred}\,(C_A, O_A), \mathsf{disjointWith}\,(O_A, O_B, St_i), \mathsf{coalitionState}\,(St_i).$$
$$\mathsf{sem\_cred}\,(C_A, O_B) \leftarrow \mathsf{sem\_cred}\,(C_A, O_A), \mathsf{subClassOf}\,(O_A, O_B, St_i), \mathsf{coalitionState}\,(St_i).$$
$$\mathsf{sem\_cred}\,(C_A, O_B) \leftarrow \mathsf{sem\_cred}\,(C_A, O_A), \mathsf{equivalentClass}\,(O_A, O_B, St_i), \mathsf{coalitionState}\,(St_i).$$
$$\mathsf{final\_sem\_cred}\,(C_A, O_A) \leftarrow \mathsf{sem\_cred}\,(C_A, O_A), \mathit{not}\ \mathsf{disj\_sem\_cred}\,(C_A, O_A).$$

The first two rules define (limit) the symmetry property of equivalence and disjoint context relations within a given state. The next three rules define semantic credential entailments in the cases of disjoint, equivalent and subclass semantic relations, based on the information of a current coalition state. We assume that the coalition state information is dynamically added to the coalition relations based on some event triggering mechanisms. The last rule entails all coalition semantic credentials being not in a conflicting semantic assignment.

Implementing semantic interoperability. Currently, most authorization approaches are based on locally issued credentials (attributes or privileges) bound to user identities. When these schemas are applied to open distributed settings they result in limited and inconvenient credential management with a lack of interoperability of attributes. It is unlikely to expect that different heterogeneous systems would unify a common homogeneous set of authorization criteria.

The approach in [9, 10] introduces an extension to a policy definition model to address potential semantic interoperability of credentials issued from different CAs. The notion of Source Of Authorization Description (SOAD) defines what third-party CA's roles (certifiable attributes) qualify to CA's own roles/attributes used to certify own users. The foundation of this approach is that trust in CA authority to certify users with certain attributes implies trust in that CA to define bilateral semantic implications of how other CAs attributes relate with the CA's own ones.

A coalition platform can adopt a SOAD-based metadata representation for high-level specification of coalition semantic context relations between partners. The benefit of adopting a SOAD-based coalition semantic relations is that it allows partners to express additional constraints on potential credential implications, such as, allow transitivity of credential implications between more than one partners, or define constraints on what attributes implications are allowed via a given semantic relation, or define multi-context semantic implications. For example a SOAD of a Blockbuster partner could state:

$$\text{Lufthansa.eID} \wedge \text{AVIS.driver}\ \texttt{subClassOf}\ \text{Blockbuster.adult}$$

meaning that if an entity has a credential C_1 coupled to a Lufthansa's semantic context (class) of electronic identity, and a credential C_2 coupled to an AVIS's semantic context of a driver, the rule will imply that entity as having any credential (e.g., C_3) in the Blockbuster's semantic access policy coupled to the Blockbuster's semantic context of adult (e.g., $\mathsf{sem\_cred}\,(C_3, \text{Blockbuster}.\mathit{adult})$). In the example we assume that Lufthansa, AVIS and Blockbuster are partners in a coalition formation.

A research direction is to ground the SOAD metadata approach, with a suitable transformation process, in a datalog representation for policy reasoning and evaluation of semantic interoperability of credentials.

3.4 Access decision process and evaluation

This section presents insights of how an access decision process is defined based on logic programming formalization. An important aspects here is to show the feasibility of the proposed access control model, and how access decision making scales with an increasing number of coalition partners.

We first define the logic level access decision process based on semantic interoperability of credentials. We define a semantic context assignment policy and a semantic interoperability policy in order to introduce an access control process handling semantic interoperability of credentials in a coalition instance.

Definition 4. (Semantic Assignment Policy) *Let $\mathcal{X}$ be a coalition instance defined by a set of partners and let $\mathcal{P}^i_{\mathcal{SA}}$ be partner i's semantic access policy. Partner i's credential semantic assignment policy $\mathcal{P}^i_{\mathcal{C}}$ is defined by the following rules generation: for any* $\mathsf{sem\_cred}\,(C, O)$ *in $\mathcal{P}^i_{\mathcal{SA}}$ generate the rule* $\mathsf{sem\_cred}\,(C, O) \leftarrow \mathsf{cred}\,(C)$.

A coalition semantic assignment policy, $\mathcal{P}_{\mathcal{C}}$, is defined by the union of all partners' semantic assignments, i.e. $\mathcal{P}_{\mathcal{C}} = \bigcup_{i \in \mathcal{X}} \mathcal{P}^i_{\mathcal{C}}$.

Client's set of credentials have syntactic values that obtain semantic values by assigning contexts to them. $\mathcal{P}_{\mathcal{C}}$ assigns semantics to any credential whose values match the values (tuple) C defined by a partner. A credential can be assigned to several semantic contexts across multiple partners' policies in a coalition, or even assigned multiple contexts within a single partner's policy.

Definition 5. (Semantic Interoperability Policy) *Let $\mathcal{X}$ be a coalition instance defined by a set of partners, $\mathcal{R}$ be the set of coalition ontology relations, $\mathcal{I}$ be the set of credential inference rules (1)–(6) shown in Figure 3, and let $\mathcal{C}^i_{\mathcal{P}_{\mathcal{SA}}}$ denote the set of all semantic credential terms occurring in partner i's semantic access policy. To face the computation of semantically equivalent credentials we add the following two rules to the structure of $\mathcal{I}$:*

$$\mathsf{equiv\_sem\_cred}\left(C', O\right) \leftarrow \mathsf{given\_sem\_cred}\,(C, O), \mathsf{final\_sem\_cred}\left(C', O\right), C \neq C'.$$

$$\mathsf{equiv\_sem\_cred}\left(C', O'\right) \leftarrow \mathsf{given\_sem\_cred}\,(C, O), \mathsf{final\_sem\_cred}\left(C, O'\right), \mathsf{final\_sem\_cred}\left(C', O'\right), C \neq C', O \neq O'.$$

A coalition semantic credential interoperability policy, $\mathcal{P}_{\mathcal{SI}}$, is defined by: $\mathcal{P}_{\mathcal{SI}} = (\bigcup_{i \in \mathcal{X}} \mathcal{C}^i_{\mathcal{P}_{\mathcal{SA}}}) \cup \mathcal{R} \cup \mathcal{I}$.

The first rule states that for a given credential term C in a semantic context O the rule entails all (different) credential terms C' assigned under the same context O. In other words, the rule entails all credentials semantically equivalent to the given one under the given context.

The second rule states that for a given credential term C in a semantic context O the rule will entail all credentials C' assigned to a context O' which the given credential is also assigned to. In that way, the rule entails all credentials semantically equivalent to the given one but under different semantic contexts.

With the new rules, policy $\mathcal{P}_{\mathcal{SI}}$ entails all (final) credential terms according to the coalition semantic relations and, out of those terms, when a given credential is added the policy entails all other equivalent terms.

The assignment policy and the semantic interoperability policy are instantiated (formed) during the coalition formation phase, in a step after partners selection took place. The access decision process described below takes place during a coalition operational phase.

Definition 6. (Logic Access Decision) *Let P be a logic program and L be a positive literal. We say that P grants access to L iff*

(i) P is logically consistent, $P \not\models \bot$ and

(ii) L is a logical consequence of P, $P \models L$.

Below we summarize the notations used in the formalization of the access decision process.

- r denotes a request for a resource.
- c denotes a credential term $\texttt{cred}(C)$.
- c_s denotes a semantic credential term $\texttt{sem\_cred}(C, O)$.
- $\hat{c}_s$ denotes a predicate symbol for a given semantic credential term. When the new symbol notation is applied on a credential set, all elements of the credential set are termed analogously. We used $\texttt{given\_sem\_cred}(C, O)$ in Definition 5.
- $\tilde{c}_s$ denotes a predicate symbol for an equivalent semantic credential term. We used $\texttt{equiv\_sem\_cred}(C, O)$ in Definition 5.
- $\mathcal{C}_{\mathcal{A}}$ denotes a client's set of active credentials presented, symbolically $\mathcal{C}_{\mathcal{A}} = \{c_1, \ldots, c_n\},\ n \geq 0$.
- $\mathcal{C}_{\mathcal{SA}}$ denotes the client's set of active credentials assigned to coalition semantic contexts, symbolically $\mathcal{C}_{\mathcal{SA}} = \{c_{s_1}, \ldots, c_{s_m}\},\ m \geq n,\ n = |\mathcal{C}_{\mathcal{A}}|$. A credential term could be assigned to more than one semantic contexts, or if not recognized in any semantic context, the credential is assigned its inherent semantics (refer to Example 1).
- $\mathcal{C}_{\mathcal{SE}}$ denotes a set of semantic credential terms semantically equivalent to those of $\mathcal{C}_{\mathcal{SA}}$ according to a coalition semantic interoperability policy, $\mathcal{P}_{\mathcal{SI}}$, symbolically $\mathcal{C}_{\mathcal{SE}} = \{c_{s_1}, \ldots, c_{s_n}\},\ n \geq 0$.
- $\mathcal{C}_{\mathcal{P}_{\mathcal{SA}}}$ denotes all semantic credential terms appearing in a semantic access policy, $\mathcal{P}_{\mathcal{SA}}$, symbolically $\mathcal{C}_{\mathcal{P}_{\mathcal{SA}}} = \{c_{s_1}, \ldots, c_{s_n}\},\ n \geq 0$.

$\texttt{SemanticAccessControl}(r, \mathcal{C}_{\mathcal{A}})$:

1. $\mathcal{C}_{\mathcal{SA}} = \{c_s \mid \mathcal{P}_{\mathcal{C}} \cup \mathcal{C}_{\mathcal{A}} \models c_s\}$;
2. $\mathcal{C}_{\mathcal{SE}} = \{c_s \mid \mathcal{P}_{\mathcal{SI}} \cup \hat{\mathcal{C}}_{\mathcal{SA}} \models \tilde{c}_s\}$;
3. $\mathcal{C}_{\mathcal{SA}} = \mathcal{C}_{\mathcal{SA}} \cap \mathcal{C}_{\mathcal{P}_{\mathcal{SA}}}$;
4. $\mathcal{C}_{\mathcal{SE}} = \mathcal{C}_{\mathcal{SE}} \cap \mathcal{C}_{\mathcal{P}_{\mathcal{SA}}}$;
5. if $\mathcal{P}_{\mathcal{SA}} \cup \mathcal{C}_{\mathcal{SA}} \cup \mathcal{C}_{\mathcal{SE}} \models r$ and $\mathcal{P}_{\mathcal{SA}} \cup \mathcal{C}_{\mathcal{SA}} \cup \mathcal{C}_{\mathcal{SE}} \not\models \bot$ then
6. *grant* access to r else *deny* access to r.

Fig. 4. Semantic access decision process

Figure 4 shows the coalition access control process. Input to the algorithm is a request r and client's set of active credentials $\mathcal{C}_{\mathcal{A}}$. Step 1 of the algorithm assigns semantic contexts to the set of active credentials according to $\mathcal{P}_{\mathcal{C}}$. Step 2 computes the set of semantic equivalent credentials to $\mathcal{C}_{\mathcal{SA}}$ according to $\mathcal{P}_{\mathcal{SI}}$. Technically, given $\hat{\mathcal{C}}_{\mathcal{SA}}$ this step computes all credential terms $\tilde{c}_s$ that are semantically equivalent to the credentials in $\hat{\mathcal{C}}_{\mathcal{SA}}$ and out of the resulting set it removes all prefixes "$\texttt{equiv\_}$" to obtain the desired set $\mathcal{C}_{\mathcal{SE}}$. Thus, step 1 assigns semantics to credentials based on all partners' semantic assignments so that step 2 defines semantic implications of $\mathcal{C}_{\mathcal{SA}}$ across all partners' relations. Steps 3 and 4 select those semantic credentials out of $\mathcal{C}_{\mathcal{SA}}$ and $\mathcal{C}_{\mathcal{SE}}$ that are relevant for the access decision step, i.e. those occurring in $\mathcal{P}_{\mathcal{SA}}$. Step 5 checks the logic access decision, i.e. if client's active credentials, $\mathcal{C}_{\mathcal{SA}}$, and their semantic implications, $\mathcal{C}_{\mathcal{SE}}$, preserve $\mathcal{P}_{\mathcal{SA}}$ consistent and satisfy the requirements for r.

Here the $\mathcal{P}_{\mathcal{SA}}$ is the semantic access policy of the partner providing the requested resource, in case, a coalition service is provided by a single coalition partner. However, if a coalition service is provided as a composition of several individual partners' services then steps 3, 4 and 5 are processed as $\mathcal{P}_{\mathcal{SA}}$ is the resulting policy of the composition of individual partners' $\mathcal{P}_{\mathcal{SA}}$. In this case, $\mathcal{C}_{\mathcal{P}_{\mathcal{SA}}}$ indicates all semantic credential terms occurring in the composite $\mathcal{P}_{\mathcal{SA}}$. Policy composition has its inherent issues [31], such as preferences of integration and resolving policy conflicts, which a partner indicates them along with the registration of its semantic access policy.

Appendix A shows in details the access control process in a scenario of a coalition formation with three partners. The proposed scenario is the basis of the following evaluation that has been performed.

Evaluation of logic level access decision. An efficient implementation of the proposed access control process has been done by using Answer Set Programming (ASP) solvers [32]. Nowadays, ASP solvers are very efficient tools that compute results even with several thousands of atoms and rules[5]. This makes them suitable as a back-end engine for the logic computations in steps 1, 2, and 5 of the decision process.

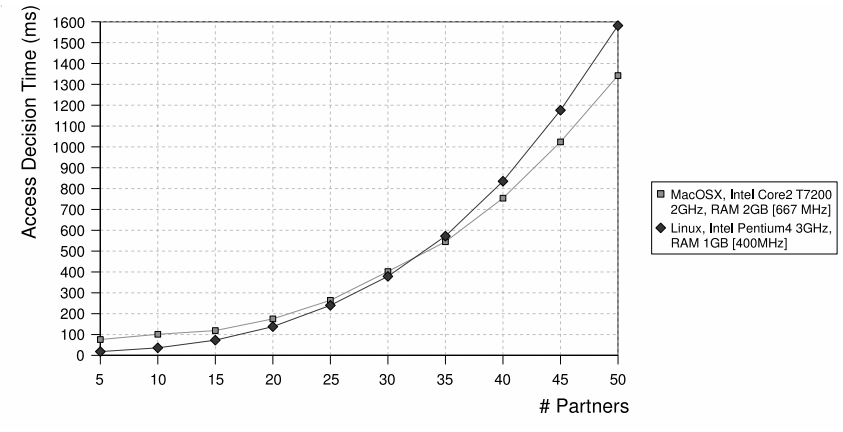


Fig. 5. Access decision evaluation

Figure 5 shows the summary of the evaluation assessment. We used DLV solver (ref. [32]) and its datalog front-end for the logic computations of deduction and consistency check. We have done a Java wrapper to DLV that implements the decision process and handles external invocations to the solver.

We followed the example scenario of Appendix A but with increased number of partners, services and credentials. We have generated data sets for 50 partners where each partner shares 10 services protected by 10 different credentials. We defined 10 semantic contexts per partner each coupled with a single credential. We related each partner's contexts with next partner's contexts following partners enumeration. In this way the more partners we included in the tests the more semantic relations were chained by the interoperability policy. We performed 10 experiments starting with 5 partners and increasing with 5 more on each next experiment. Each experiment reports an average access decision time of 10 trials.

We performed the experiments on two different PC configurations to compare their performance. In case of 50 partners, resulting with 500 credential atoms and 494[6] coalition relations in the interoperability policy, and 500 rules in the coalition semantic assignment policy, the measured access decision time was around 1.6 seconds. Although a restriction on decision time is subjective to a coalition type and a business model, we estimated that up to 45 partners in a coalition *instance* limits the threshold of efficient access decision (less than a second). The threshold estimation is subject to the conditions above of average number of credentials and semantic contexts per partner.

4 Automated Credential Establishment with Coalition End-Users

In the context of HDC formations, end-users are unaware of current coalition configuration and what set of credential requirements are necessary to access a service of a coalition instance. An automated credential negotiation process is envisaged to apply on each step of a coalition execution where a coalition server negotiates with an end-user current service's requirements.

The work in [33] proposes an interactive access control model (IAC) where a server interacts with a client requesting it for missing credentials necessary to grant a resource. The client, on its turn, checks

[5] ASP benchmarks available at http://asparagus.cs.uni-potsdam.de

[6] We kept the first three partners' coalition relations as in Appendix A.

if it has the requested credentials and sends a response back to the server. The server re-evaluates an access policy to verify if the returned set of credentials grants access to a requested service. In case the client does not have all credentials from the first round, the server re-computes a new set of missing credentials and asks them to the client. A client and a server interact until either the client presents a set of credentials satisfying server's access policy, or there is no missing set to be asked to the client and the server denies access.

An extension to IAC [34] empowers a client entity as having its own IAC reasoning so that whenever a server asks a client for a set of missing credentials the client computes, according to its own credential control policy, what missing credentials the server has to present to see the client's credentials. The extended model defines a negotiation protocol, on top of IAC model, allowing a client and a server to interact until an agreement is reached and the server provides access to the requested object, or one of the parties denies the negotiation process.

There are several other trust negotiation approaches such as TrustBuilder [35], Trust-X [36], PeerTrust [37] that one might consider adopting for an interactive access establishment process between an end-user and a coalition server. Important for an automated negotiation process are the security policies behind. In a negotiation system one can find a security (access) policy encapsulating access control statements, and a disclosure policy encapsulating credential disclosure control statements. The disclosure policy is either embedded in the security policy as a meta-policy, or provided as a separate policy specification.

Each partner along with its semantic access policy supplies the necessary additional policies for an automated negotiation process instantiated by means of the coalition-wide semantic policy vocabulary. In this way, partner's security policies are transformed into internal representation, in our case in a logic program format, for subsequent enforcement by the coalition platform.

Automated credential negotiation with semantic interoperability. An important aspect regarding a coalition usage is the ability of end-users to establish necessary access rights with a coalition platform server using the coalition-specific semantic interoperability of credentials. A research challenge here is to customize a negotiation process allowing for on the fly evaluation of credential interoperability implications achieving a smooth and automated credential agreement. Essentially, during a negotiation process initialization phase, or on each credential exchange request, a platform negotiation server will provide an end-user with a current credential interoperability policy. A coalition credential interoperability policy is derived based on a current coalition state and coalition partners configuration.

The necessity of sending an interoperability policy to an opponent agent is to minimize the possible expansion of credential requests to the opponent reflected by the coalition credential interoperability, and to give the end-user agent to compute what credentials among those available to the user are equivalent to the requested ones. The following example illustrates the envisaged interaction process.

Example 3. Let Europcar and Blockbuster are partners in a given coalition formation and let Blockbuster semantic access policy define access to rent-a-dvd service with a restricted movie category by a Blockbuster's adult membership certificate issued to clients who have proved their over 18 status.

$$\texttt{grant}(rent\_a\_dvd, restricted) \leftarrow \texttt{sem\_cred}(adult\_membership, over18).$$

Let Europcar and Blockbuster have agreed on the following coalition semantic relation:

$$\texttt{subClassOf}(Europcar.driver, Blockbuster.over18)$$

and have provided their respective semantic credential assignments:

$$\texttt{sem\_cred}(adult\_membership, over18) \leftarrow \texttt{cred}(adult\_membership).$$
$$\texttt{sem\_cred}(driving\_license, driver) \leftarrow \texttt{cred}(driving\_license).$$

The first assignment rule states that if a credential with an attribute adult_membership is presented then the credential is assigned a semantic context of over18. Analogously, if a driving license is presented, it is assigned a semantic context of a driver.

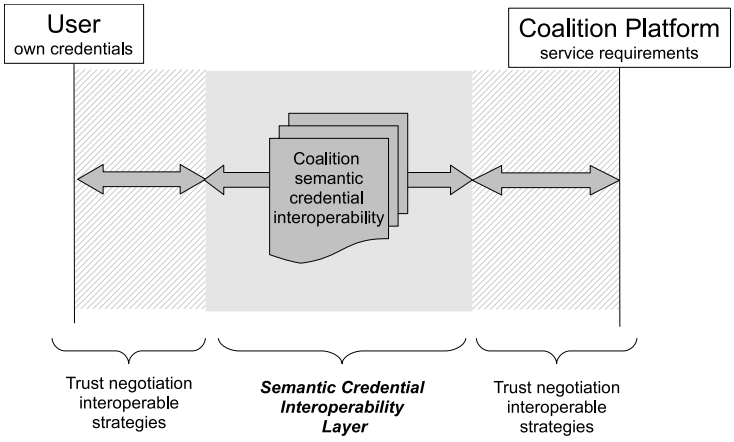


Fig. 6. Automated trust negotiation with semantics of credential interoperability

Now, if a user with a driving license certificate wants to rent a dvd movie in a restricted category via the Blockbuster's own online system and he does not have the Blockbuster's adult membership certificate, the same will be denied access to the service.

If the user uses the coalition platform facilities to access the rent-a-dvd service of Blockbuster, a negotiation process will request the user to present its Blockbuster's adult membership certificate along with a current coalition interoperability policy including the above semantic relation and credential assignments.

The user will evaluate that he does not have the adult membership certificate, but he has a driving license certificate that semantically implies the requested one, and presents it to the platform server. The coalition server, in turn, will verify the driving license certificate according to the trusted authorities, as specified by the Europcar during a registration phase, and then will compute, similarly to the client side, the semantic credential implication of an adult membership certificate. Next, the server will evaluate if client's presented credentials together with those out of the semantic interoperability implications satisfy the service requirements, and will grant access to the rent-a-dvd service.

There is also an issue of possible expansion of certification authorities supported by a coalition, for example, Europcar could maintain a set of respective (governmental) CAs of European countries responsible for issuing driving license certificates. Such information of potential CAs could be included in the semantic interoperability policy, so that, a user can validate if its driving license certificate is among those supported by a given coalition.

A research direction is the integration of the concept of semantic credential interoperability into iAccess[7], our prototype system for automated trust negotiation. iAccess negotiation model is built on two logic reasoning services: deduction and abduction [38]. The main integration aspect is the use of abduction reasoning for the computation of a set of credentials out of the client's available ones that satisfy a given requested credential according to an interoperability policy. In this way, the user's negotiation agent on receiving a credential request by a coalition platform, will compute a set of own credentials semantically equivalent to the requested one, and will continue the negotiation process for the set of own credentials. The abduction in this context will provide a feasible computation of semantically equivalent credentials because, first, the set of hypotheses to the abduction reasoning is limited to the number of user's own credentials and, second, the interoperability policy is a stratified logic program as already discussed in Section 3.3. User's own credentials are expected to be much fewer than the potential number of semantically equivalent credentials defined by a coalition formation.

[7] http://www.interactiveaccess.org

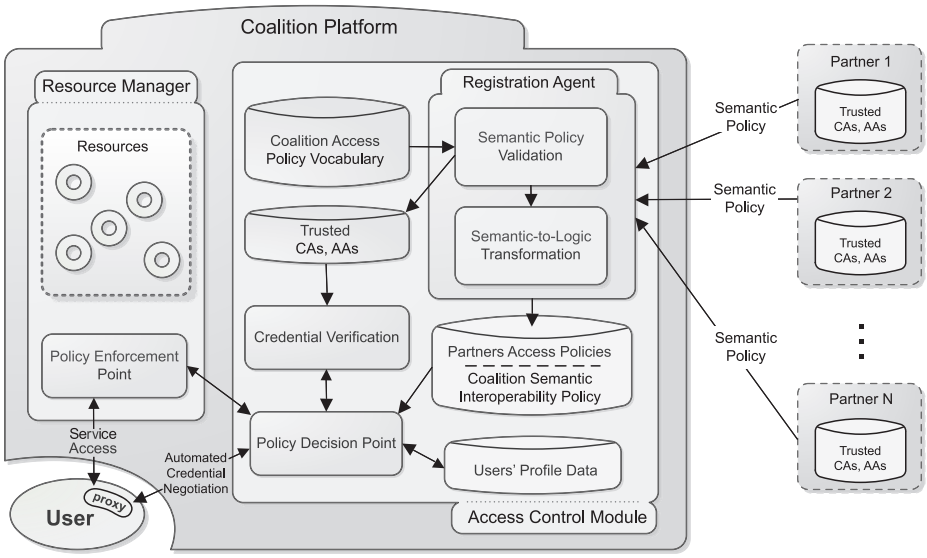


Fig. 7. Coalition access control module architecture

Finally, an entity can use many different strategies to negotiate trust, such as willingness to succeed with suspicious or eager modes, length of negotiation, amount of credential disclosure, privacy preserving strategies, computational effort expended, to name a few. There is a dedicated research direction on interoperable strategies for automated trust negotiation [39–41] focusing on defining classifications of family of strategies that interoperate, i.e., any two chosen strategies under a given family will guarantee successful negotiation whenever possible – whenever existing mutually satisfiable credential requirements.

Our approach on semantic interoperability of credentials complements the strategies interoperation. We do so by adding an additional interoperability layer on top, that allows strategies to succeed not only whenever mutually satisfiable credential requirements are explicitly expressed, but also discovering existing implicit mutually satisfiable requirements based on the semantics of credential usage and coalition interoperability. Figure 6 shows the newly identified semantic interoperability layer of automated credential negotiation.

5 Coalition Platform Access Control Architecture

The proposed architecture summarizes the identified access control issues for HDCs and gives an overall view of the necessary components supporting the presented semantic access control model. Figure 7 shows the architecture of a coalition access control module and its interactions with the coalition resource manager and coalition end-users.

The access control module has two main parts: a registration agent and a policy decision point (PDP). The registration agent assists partners in registering resources to the platform, and in registering their semantic access policies protecting the resources. The registration agent validates if partners' semantic policies conform to the coalition-wide semantic policy description (vocabulary), and then transforms them to internal logic representation (as discussed in Section 3.2).

Part of the semantic policy information, each partner also provides the public-key certificates of trusted certification authorities for the credentials used in the semantic policy. When the registration agent validates the semantic policy it extracts and stores the set of trusted CAs to an internal database. There is a credential verification module that verifies and validates digital certificates based on partners' trusted CAs. The PDP consults the credential verification module when it needs to verify user's credentials.

Users interact with a coalition resource manager (in case of a business process, a BPEL manager) for accessing coalition services. Users' interactions are via Web services invocation mechanisms and the corresponding transport protocols. To facilitate transparent credential negotiation between an end user and a coalition server (a platform), we adopt the concept of a user-side proxy component that faces potential negotiations with the coalition server (precisely, with the PDP). To achieve that, the proxy component has to handle all service invocations made by the user, and for those where more access rights are necessary, the proxy interacts with the PDP to establish these rights. Once access rights are established, the proxy stays active for potential next interactions during the coalition service execution. The last is the case when the invoked service is a composition of partners' services.

There is a policy enforcement point (PEP) component part of the resource manager that consults the PDP for access decisions, and enforces the decisions accordingly. The PEP interacts with the PDP on each service execution by the resource manager, and for those execution steps where more credentials are necessary, the PEP suspends the resource manager and informs the user-side proxy on that. The proxy on its turn connects again with the PDP for establishing the necessary credentials.

The above described interactions are influenced by the fact that users might often be behind a firewall network, and any direct call-back form the PDP to the user proxy may be denied by the firewall service. To avoid that, on handling a service invocation, the proxy awaits on the PEP for a notification (as a regular service response) if more access rights are necessary, and then initiates a credential negotiation request with the PDP. When the negotiation process is over, the proxy invokes the original service with the result of the negotiation, and awaits active for further notifications. The PEP, on its turn, either resumes the service execution (in case of sufficient access rights), or aborts the service execution (if insufficient rights).

Since there could be multiple coalition instances active, the PEP sends to the PDP all information necessary (e.g., coalition instance identifier, service identifier, action, etc) for the PDP to allocate the corresponding access control policy of the service, and the corresponding coalition specific policies, such as the assignment policy and the interoperability policy. We note that during the coalition formation phase, all coalition specific policies are derived from the coalition model and partners' profiles, and made available for evaluation during operation.

The PEP and the PDP may reside on a same physical location (but with different service APIs) or on different locations depending on a specific platform design. In the case of different physical locations, important factor to be considered is the efficiency (overhead) of calls between the PEP and the PDP, and an additional level of security for the interactions between them.

The PDP implements the access decision process described in Section 3.4, and the negotiation schema of [34] (with its interactive access control process [33]) on top of the decision process. The user-side proxy functionality could be provided either as a pre-installed software agent, or as a specific coalition service (with an appropriate GUI) configurable by end-users.

Our main goal regarding negotiation is to reduce the number of potential interactions. We achieve this by keeping a user's profile of active credentials, so that the underlying semantic interoperability process can take an automated decision on subsequent service requests by the user. There is a data set that keeps information about user profiles. It contains information, such as, user's set of active credentials presented to a coalition platform, credential information of what a user has declined to present, and so on. Whether this data set stores information only of a current session or of user's past interactions depends on specifics of the coalition. To this extend, we assume that a coalition platform has a predefined policy of how to maintain user information.

There are also other important architectural elements of the platform, complementing those of the access control module, that one has to consider when designing a complete platform architecture. The work in [42] provides a good overview on those.

6 Related Work

We overview existing approaches to semantics of access control for virtual organizations and dynamic coalitions. We highlight the suitability of some of them to our needs (identified in Section 2). Then, we

give an overview of technology standards for HDC business modelling, and discuss relevant approaches to definition of life cycles of coalitions.

We will start with the EU project TrustCoM[8] which has developed a comprehensive VO management life cycle [43], and an environment for trust and security in B2B collaborations. The project has elaborated trust and security solutions [2, 42, 5–8] for evolving (dynamic) VOs in terms of membership and structure. The security solutions and tools for implementing contractual terms and policies elaborated within the project can be well applied in the context of HDCs. However, there is a difference in deploying these solutions that comes from the difference of VO life cycle design between HDCs and TrustCoM's VOs. Section 6.3 discusses the details on the difference of the two VO life cycles. The advocated access control model in the paper adds a new layer to the TrustCoM's security infrastructure where it provides an interoperable access control process to VO-shared services.

The study of policy management for coalition operations is a multifaceted field tightly related to the policy management of Virtual Organizations. The problem of defining a policy for governing VO operations composed by several partners is faced in [44]. It uses a VO-wide operational policy along with VO policy on resources and VO policy on users. The approach does not allow mapping between partners' local policy and VO's policy. Overcoming the previous limitation, [45] proposes a Trust-based Access Control combining global and local trust relationships among VO's parties. In general, the use of trust-levels is better suited for long-term VO's relationships, rather than for the highly dynamic coalitions we are interested in.

Inter-organizational access control for VOs is studied in [46], extending RBAC with Role Based and Mapping Access Control to allow VO users to request resources in a VO they do not belong to. Assuming that each trust domain controls its own policies independently (using Local Roles), the use of a Global Role (common to a VO) is proposed as an interface among Local Roles. An extension of RBAC is presented in [47] permitting the expression of contextual permissions and the abstraction of the different entities (users, actions and roles) in order to separate policies from the dynamic infrastructure. However, the former approach loses efficiency as the number of VOs increases, while the latter proposes an explicit abstraction process that is not appropriate for dynamic coalitions.

During the coalition operation phase conflicting policies may occur especially when partners' services interact with each other, for example as part of a service composition process, and the resulting policy composition may require special reconciliation rules for proper policy enforcement. A recent work [31] proposes an extension to XACML [48] policy integration algorithms to handle policy reconciliation in dynamic and open environments.

6.1 Semantic access control approaches

The standardization efforts of W3C [15] on extending the World Wide Web with semantics have opened a new direction of automated machine processing of data to support different needs and services. A number of works have been presented that apply semantics to enhance authorization interoperation across semantically heterogeneous systems [49, 50, 10].

Warner et al [49] propose a framework for participants of an organization to gain access to organizations' resources in a coalition environment with syntactically and schematically heterogeneous policies. They propose a model and an algorithm that define what attributes are relevant for an internal role(s) in an organization so that an external user having these attributes is granted access to the resource. The authors use a hierarchy of concepts to categorize different attributes used in that organization and validates if a given set of external attributes matches those required for an internal user to access a local resource. Our model complements [49] by defining semantic relations between organizations' internal concept in a coalition.

The work in [50] proposes a model that uses organizations' ontologies to achieve information interoperability by defining two sets of relations: one for interlinking roles between organizations and another for interlinking (grouping) objects under semantic equivalent concepts. In this model a user has permission

[8] http://www.eu-trustcom.com

to a remote object if (i) user's local role has the same permission to a local object; (ii) the local object is semantically equivalent to the remote one; and (iii) the user's local role is semantically equivalent to a remote role which has the same permission on the remote object (the remote role and object are internal to an organization). Their approach is orthogonal to ours, we define interoperability of credential semantics on coalition level and provide access to resources based on semantics of credentials and not on semantic relations between resources.

A number of research efforts have targeted the RBAC model [27] using semantic Web technologies, such as implementation of RBAC as a service [51], or expressing RBAC constraints [52], or expressing RBAC with negative authorizations [53], or expressing RBAC using the OWL language [26]. As discussed in Section 3.2, the work in [26] provides a comprehensive treatment of defining RBAC models based on OWL and discusses the foundations and necessity of semantically modelling more general models of attribute based access control [54] and usage control [55].

There are several works dedicated to developing Semantic Web-based languages [56, 57, 25], i.e. languages that consider the semantics of policies, and allowing access policies to be described across domains with heterogeneous information models facilitating common understanding among entities. Another approach [58] applies the Semantic Web for addressing users' privacy concerns in automated trust negotiations. A comparison study of suitability of recent policy languages for automated trust negotiation in the context of security and privacy protection on the Semantic Web is discussed in [59].

Defining access policies of HDCs. The Semantic Policy Language (SPL) part of the Semantic Access Control (SAC) model [9, 10] provides a good foundation for the definition of access control policies for HDCs. The definition of such policies is a complex activity that presents many similarities with computer programming. Consequently, SPL includes a same type of mechanisms used for reducing the complexity in computer programming languages, such as modularity, parametrization and abstraction. The ability to define modular policies constitutes the basis of the solution to (i) defining a generic coalition policy before knowing the partners' policies and (ii) automatically deriving a coalition-specific policy on the basis of the generic coalition policy and the policies of the specific partners involved in an instance of the coalition. Additionally, it is necessary that a coalition policy takes into account, not only on the partners' policies, but also context conditions. In SPL, the use of semantic information about the context allows the administrator to include contextual considerations in a transparent way.

Usual components of access policies include the target resource, the conditions under which access is granted/denied and, sometimes, access restrictions. As opposed to other languages, specifications in SPL do not include references to the target object. Instead, a separate specification called Policy Applicability Specification (PAS) is used to relate policies to objects dynamically at the time of policy evaluation. The PAS provides an expressive way to relate policies to resources, either explicitly or based on the metadata about the objects (e.g. type of content, owner, price, etc.). PAS specifications include three main elements: policy, objects and instantiation. The policy element indicates which policy is applicable to the specified objects. In our case, the objects are the services offered and used by the coalition, which in turn expose different operations. These objects are defined specifying their location and conditions expressed in terms of their semantics. Operation elements are used to define which operations of the target object are controlled by the declared policy, allowing a finer grained access control. In case no operation element is included, the policy is considered to be applicable to all of the object operations. The instantiation element describes the mechanism to instantiate parameters in the policies.

Both, SPL policies and PAS use semantic information about the controlled services, and other contextual information. SPL policies and PAS can be parametrized allowing the definition of flexible and general policies, thus reducing the number of different policies to manage. Parameters, which can refer to complex XML elements, are instantiated dynamically from semantic and contextual information during the coalition formation phase. Moreover, policies can be composed importing components of other policies without ambiguity. This compositional approach allows us to define the abstract meaning of the elements of the policies, providing a mechanism to achieve abstraction, which also helps in reducing the complexity

of management. The schema for SPL specifications is represented as a set of XML-Schema templates that facilitate the creation of these specifications, allowing their automatic syntactic validation [9].

6.2 Technology standards for HDC business modeling

Business modeling is a key element in the coalition definition phase. Each coalition model is based on a business model. The business model is used to describe the processes, workflows and services offered by a coalition. Additionally, the definition must be abstract enough to deal with partially-defined processes because the coalition model does not include any specific process, but references to the processes provided by the partners. Therefore, as an additional requirement for the definition of coalition business models is the need to support the definition of partners' roles and services. This can be achieved with several existing languages.

Business Process Modeling Language (BPML) is an XML-based metalanguage developed by the Business Process Management Initiative[9] (BPMI) as a means of modeling business processes, much as XML is, itself, a metalanguage with the ability to model enterprise data. An associated query language, Business Process Query Language (BPQL) has been developed by Initiative members as a standard management interface that can be used to deploy and execute defined business processes. According to BPMI, BPML and BPQL will be used to establish a standardized means of managing e-business processes through Business Process Management Systems, similarly to the way that SQL established a standardized means of managing business data through packaged database management systems (DBMSs). Both BPML and BPQL are open specifications.

The eXtended Business Modeling Language[10] (XBML) is used to define the business processes of an organization. It is based upon a 5 dimensional business framework (What, Who, Where, When and Which) and is uniquely supported by approximately 55 rules that govern the usage, "output" and "syntax" of the language. XBML enables highly consistent, complete and detailed business models to be created, and provides a disciplined methodology to describe a business and its underlying processes. This language is rapidly becoming a standard "front-end" that many organizations use to define business operations. The output is business friendly, portable and can be used by many BPM applications.

Business Process Execution Language for Web Services[11] (BPEL4WS) provides a means to formally specify business processes and interaction protocols. BPEL4WS provides a language for the formal specification of business processes and business interaction protocols. By doing so, it extends the Web Services interaction model and enables it to support business transactions. BPEL4WS defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to-business spaces. More recently, the BPEL4WS has been continued in the Web Services Business Process Execution Language[12] (WSBPEL). Processes in WSBPEL export and import functionality by using Web Service interfaces exclusively. An important feature of WSBPEL is that it allows business processes to be described in two ways: (i) Executable business processes that model actual behavior of a participant in a business interaction, and (ii) Abstract business processes that are partially specified processes not intended to be executed. An abstract process may hide some of the required concrete operational details, and it serves a descriptive role with more than one possible use cases, including observable behavior and process template. In this sense, the abstract processes provide good support for our need to define business processes for HDCs.

6.3 Virtual organization life cycles

There are several works identifying VO life cycles [60, 61, 43, 62]. A reference point [60] defines a comprehensive four-stage VO life cycle and the corresponding data management activities to each phase. The

[9] http://www.bpmi.org
[10] http://www.businessgenetics.net
[11] http://www.ibm.com/developerworks/library/ws-bpel
[12] http://www.oasis-open.org/committees/wsbpel

life cycle was further elaborated in [61], within the EU research project TrustCoM, and also adopted in the Virtual Enterprise community (e.g., [63]).

We will review the VO life cycle of the TrustCoM project [43], which extends that of [60], has been adopted by another European project CoreGRID[13], and has several similarities with the VE community. The TrustCoM project covers a domain of VO research relatively close to our purposes. Each VO has an initiator (entity) who is responsible for creating and managing the VO. Next is a short description of the VO-life cycle.

Specification also called, a preparatory phase of a VO, where an initiator entity specifies the business process, required roles and message workflow of a VO.

Identification a post-phase of the specification, where additional service requirements are identified and made available to the VO specification data.

Formation the phase where VO members are looked for, selected and assigned business roles of the VO specification. Partners profiles are derived from the information in the previous two phases, and based on those profiles, a selection process takes place. There is a period of negotiation between the initiator and selected partners for agreement on the service requirements to be covered/fulfilled by the selected partners. This phase also includes distributing information such as policies, Service Level Agreements (SLAs), etc, and the binding of the selected candidate partners into the actual VO.

Operation after the formation phase, the VO is considered as ready to enter the operation phase. This phase puts in effect the VO as officially active business. The identified and properly selected VO members perform accordingly to their roles assigned. There is an explicit sub-state, called "dormant", where the VO is inactive due to some contractual exceptions to be handled. Additionally, VO membership and structure may evolve over time to adapt to new opportunities in the business environment.

Dissolution the last phase of a VO, when the objectives of the VO has been fulfilled, or some contractual violation occurs that forces VO dissolution.

The formation phase, in the above described life cycle, is instantiated N times per VO, i.e., for any member identified to provide services in the VO. The resulting negotiation sub-phase is due to the need to "customize" the binding of partners' profiles to the requirements of services that partners have to conform to during the VO operation.

In contrast, the coalition formation in our life cycle is applied once for its instance lifetime. We achieve so by adding an explicit and separate registration phase before the formation phase, where partners define and register their profiles for available business models and roles. Essentially, the negotiation and agreement part of the TrustCoM's formation phase can be seen as being allocated to the partner registration phase, so that the VO formation can be automatically and dynamically established on demand.

The dynamic notion of TrustCoM's VOs is focused on the VO operation phase where VO membership and structure may evolve in order to adapt to new market (business) conditions, while in our case, the dynamic nature of HDCs comes from the necessity of dynamic formation with a main focus on prompt reaction to some events. The difference of life cycles comes from that fact that the TrustCoM project focuses on handling more complex and long-term business interactions, while our focus is on VO business models with strong timing constraints on their formation and operation.

The operation phase of the TrustCoM's life cycle and in our model coincide, but with the difference that the explicit "dormant" state may not be feasible in our context due to the strong timing constraints posed on VO operation. While the identified evolution of VO membership and structure during operation is also identified as a desirable feature of HDC through the dynamic partner replacement challenge.

7 Conclusions and Future Work

In this paper we have analyzed the problem of access control interoperability in highly dynamic coalitions (HDCs). We refer to HDCs for a domain of coalition formations that provides rapid and dynamic response

[13] http://www.coregrid.net

to market opportunities for small and medium enterprises. We have characterized the life cycle of HDCs and analyzed the access control challenges faced in each of the phases. We have presented an access control model for interoperability of partners' access control requirements in a HDC. Below we summarize the key features of the model:

- Enhancing partners' access control requirements with semantics of their usage;
- Unified semantic representation of syntactically different partners' requirements by means of coalition-wide policy vocabulary;
- Use of standard ontology class relations to relate partners' semantics at coalition level;
- Use of logic programming as an underlying model for policy evaluation and reasoning.

We have also presented an extension to an automated trust negotiation process by adding a new credential interoperability layer on top of possible interoperable negotiation strategies. The new layer allows for discovering implicit mutually satisfiable requirements by use of semantics of credentials and their interoperation.

Future work will focus on:

- Extending the coalition-wide semantic policy vocabulary to handle more general access control models (beyond RBAC) and relevant policy composition aspects, grounded in logic programming;
- Refining the SOAD meta model in a suitable datalog formalization for sound transformation of coalition-level semantic interoperability of credentials to logic programming;
- Integrating in our prototype negotiation system the semantic interoperability of credentials for achieving efficient access control process to coalition services;
- Refining the exact protocol steps (with corresponding messages) of the interactions between the client-side proxy, the PEP and the PDP in order to handle coherent end-to-end access control enforcement, especially in case of services compositions;
- Qualitative analysis on the overall architecture and evaluation of its performance and scalability.

Acknowledgments

This work was supported by the Marie Curie Intra-European fellowship 038978-iAccess within the 6th European Community Framework Programme, and partially supported by the European project OKKAM - Enabling a Web of Entities (contract ICT-215032) and by the Spanish project DESEOS (TIC-4257, Dispositivos Electrónicos Seguros para la Educación, el Ocio y la Socialización) funded by the local government of Andalucia.

We thank Francisco Sánchez-Cid for the fruitful discussions we have had together during the preparation of the work.

We thank the anonymous reviewers for their constructive comments.

References

1. Byrd, G.T., Gong, F., Sargor, C., Smith, T.J.: Yalta: A secure collaborative space for dynamic coalitions. In: Proceedings of the IEEE Workshop on Information Assurance and Security. (2001)
2. Djordjevic, I., Dimitrakos, T., Romano, N., Randal, D.M., Ritrovato, P.: Dynamic security perimeters for inter-enterprise service integration. Future Generation Comp. Syst. **23**(4) (2007) 633–657
3. Camarinha-Matos, L.M., Silveri, I., Afsarmanesh, H., Oliveira, A.: Towards a Framework for Creation of Dynamic Virtual Organizations. In: Collaborative Networks and their Breeding Environments. Springer (2005) 69–80
4. Kafeza, I., Kafeza, E., Chiu, D.: Legal issues in agents for electronic contracting. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 5, IEEE Computer Society (2005) 134.1–10
5. Arenas, A.E., Djordjevic, I., Dimitrakos, T., Titkov, L., Claessens, J., Geuer-Pollman, C., Lupu, E., Tuptuk, N., Wesner, S., Schubert, L.: Towards Web services profiles for trust and security in virtual organisations. In: Collaborative Networks and their Breeding Environments. Springer (2005)

6. Karabulut, Y., Kerschbaum, F., Massacci, F., Robinson, P., Yautsiukhin, A.: Security and trust in IT business outsourcing: a manifesto. Electr. Notes Theor. Comput. Sci. **179** (2007) 47–58
7. Karabulut, Y.: Investigating the trust management approaches for enabling trustworthy business processing in dynamic virtual organizations. In: Proceedings of the 7th Int. Conf. On Electronic Commerce Research, IEEE Computer Society (June 2004)
8. Kerschbaum, F., Haller, J., Karabulut, Y., Robinson, P.: Pathtrust: A trust-based reputation service for virtual organization formation. In: Proceedings of the 4th International Conference on Trust Management (iTrust 2006). Volume 3986 of LNCS., Springer (2006) 193–205
9. Yagüe, M., Maña, A., López, J.: A metadata-based access control model for web services. Internet Research **15**(1) (2005) 99–116 DOI 10.1108/10662240510577095.
10. Yagüe, M., Gallardo, M., Maña, A.: Semantic access control model: A formal specification. In: Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS'05). Volume 3679 of Lecture Notes in Computer Science., Springer (2005) 24–43
11. López, J., Maña, A., Muñoz, A.: A secure and auto-configurable environment for mobile agents in ubiquitous computing scenarios. In: Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing (UIC'06), Springer (2006) 977–987
12. Weber, I., Haller, J., Mulle, J.A.: Automated derivation of executable business processes from choreographies in virtual organisations. International Journal of Business Process Integration and Management **3**(2) (2008) 85–95 DOI 10.1504/IJBPIM.2008.020972.
13. Weber, I., Markovic, I., Drumm, C.: A conceptual framework for composition in business process management. In: 10th International Conference on Business Information Systems (BIS 2007), Poznan, Poland, Springer Berlin / Heidelberg (April 2007) 54–66
14. X.509: The directory: Public-key and attribute certificate frameworks (2005) ITU-T Recommendation X.509:2005 | ISO/IEC 9594-8:2005.
15. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL web ontology language reference (February 2004) http://www.w3.org/TR/owl-ref.
16. Li, N., Mitchell, J.C.: Datalog with constraints: A foundation for trust management languages. In: Proceedings of the 5th International Symposium on Practical Aspects of Declarative Languages, Springer-Verlag (2003) 58–73
17. Li, N., Grosof, B.N., Feigenbaum, J.: Delegation logic: A logic-based approach to distributed authorization. ACM Transactions on Information and System Security (TISSEC) **6**(1) (2003) 128–171
18. Li, N., Mitchell, J.C., Winsborough, W.H.: Design of a role-based trust-management framework. In: Proceedings of IEEE Symposium on Security and Privacy, 2002. S&P, IEEE Press (2002) 114–130
19. Jim, T.: Sd3: A trust management system with certified evaluation. In: Proceedings of IEEE Symposium on Security and Privacy, IEEE Computer Society (2001) 106
20. Becker, M.Y., Sewell, P.: Cassandra: Distributed access control policies with tunable expressiveness. In: Proceedings of the 5th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY). (2004) 159–168
21. Ceri, S., Gottlob, G., Tanca, L.: What you always wanted to know about datalog (and never dared to ask). IEEE Transactions on Knowledge and Data Engineering **1**(1) (1989) 146–166
22. Apt, K.: Logic programming. In van Leeuwen, J., ed.: Handbook of Theoretical Computer Science. Elsevier (1990)
23. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In Kowalski, R., Bowen, K., eds.: Proceedings of the Fifth International Conference on Logic Programming (ICLP'88), MIT-Press (1988) 1070–1080
24. Li, N., Mitchell, J.C., Winsborough, W.H.: Design of a role-based trust-management framework. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy (SP'02), IEEE Computer Society (2002) 114
25. Tonti, G., Bradshaw, J.M., Jeffers, R., Montanari, R., Suri, N., Uszok, A.: Semantic Web languages for policy representation and reasoning: A comparison of KAoS, Rei, and Ponder. In: Proceedings of the 2nd International Semantic Web Conference (ISWC2003), Springer Berlin/Heidelberg (2003) 419–437
26. Finin, T., Joshi, A., Kagal, L., Niu, J., Sandhu, R., Winsborough, W., Thuraisingham, B.: ROWLBAC: representing role based access control in OWL. In: Proceedings of the 13th ACM symposium on Access control models and technologies (SACMAT'08), ACM (2008) 73–82
27. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Trans. Inf. Syst. Secur. **4**(3) (2001) 224–274

28. Sandhu, R., Coyne, E., Feinstein, H., Youman, C.: Role-based access control models. IEEE Computer **39**(2) (February 1996) 38–47
29. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The description logic handbook: theory, implementation, and applications. Cambridge University Press (2003)
30. de Bruijn, J., Lara, R., Polleres, A., Fensel, D.: OWL DL vs. OWL Flight: conceptual modeling and reasoning for the semantic Web. In: Proceedings of the 14th International Conference on World Wide Web (WWW'05), ACM (2005) 623–632
31. Mazzoleni, P., Crispo, B., Sivasubramanian, S., Bertino, E.: XACML policy integration algorithms. ACM Transactions on Information and System Security **11**(1) (2008) 1–29
32. ASP Solvers: Some of the most known ASP solvers – *Cmodels*: http://www.cs.utexas.edu/∼tag/cmodels; *aspps*: http://www.cs.uky.edu/ai/aspps; *DLV*: http://www.dbai.tuwien.ac.at/proj/dlv; *Smodels*: http://www.tcs.hut.fi/Software/smodels.
33. Koshutanski, H., Massacci, F.: Interactive access control for autonomic systems: from theory to implementation. ACM Transactions on Autonomous and Adaptive Systems (TAAS) **3**(3) (August 2008)
34. Koshutanski, H., Massacci, F.: A negotiation scheme for access rights establishment in autonomic communication. Journal of Network and System Management **15**(1) (March 2007)
35. Winslett, M., Yu, T., Seamons, K., Hess, A., Jacobson, J., Jarvis, R., Smith, B., Yu, L.: Negotiating trust in the Web. IEEE Internet Computing **6**(6) (Nov/Dec 2002) 30–37
36. Bertino, E., Ferrari, E., Squicciarini, A.C.: Trust-X: A peer-to-peer framework for trust establishment. IEEE Transactions on Knowledge and Data Engineering **16**(7) (2004) 827–842
37. Nejdl, W., Olmedilla, D., Winslett, M.: PeerTrust: Automated trust negotiation for peers on the semantic web. In: VLDB Workshop on Secure Data Management (SDM). Volume 3178 of Lecture Notes in Computer Science., Springer (August 2004) 118–132
38. Shanahan, M.: Prediction is deduction but explanation is abduction. In: Proceedings of IJCAI'89, Morgan Kaufmann (1989) 1055–1060
39. Yu, T., Winslett, M., Seamons, K.E.: Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation. ACM Transactions on Information and System Security (TISSEC) **6**(1) (2003) 1–42
40. Squicciarini, A., Bertino, E., Ferrari, E., Paci, F., Thuraisingham, B.: PP-trust-X: A system for privacy preserving trust negotiations. ACM Transactions on Information and System Security **10**(3) (2007) 12
41. Baselice, S., Bonatti, P.A., Faella, M.: On interoperable trust negotiation strategies. In: Proceedings of IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'07), IEEE Computer Society (June 2007) 39–50
42. Dimitrakos, T., Laria, G., Djordjevic, I., Romano, N., D'Andria, F., Trpkovski, V., Kearney, P., Gaeta, M., Ritrovato, P., Schubert, L., Serhan, B., Titkov, L., Wesner, S.: Towards a grid platform enabling dynamic virtual organisations for business applications. In: Proceedings of the 3rd Third International Conference on Trust Management (iTrust'05). Volume 3477 of LNCS., Springer (2005) 406–410
43. Robinson, P., Karabulut, Y., Haller, J.: Dynamic virtual organization management for service oriented enterprise applications. In: Proceedings of the 1st International Conference on Collaborative Computing: Networking, Applications and Worksharing, IEEE (December 2005)
44. Wasson, G., Humphrey, M.: Toward explicit policy management for virtual organizations. In: Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'03), IEEE Computer Society (2003) 173–182
45. Lin, A., Vullings, E., Dalziel, J.: A trust-based access control model for virtual organizations. In: Fifth International Conference on Grid and Cooperative Computing Workshops (GCC'06), Los Alamitos, CA, USA, IEEE Computer Society (2006) 557–564
46. Che, B., Yang, G.: Research on cross-realm resource access control based on virtual organizations. In: Proceedings of the 1st International Symposium on Pervasive Computing and Applications (SPCA'06), IEEE Computer Society (2006) 222–226
47. Nasser, B., Laborde, R., Benzekri, A., Barrere, F., Kamel, M.: Access control model for inter-organizational grid virtual organizations. In: OTM Workshops. Volume 3762 of Lecture Notes in Computer Science., Springer (2005) 537–551
48. XACML: eXtensible Access Control Markup Language (XACML) (2005) www.oasis-open.org/committees/xacml.
49. Warner, J., Atluri, V., Mukkamala, R., Vaidya, J.: Using semantics for automatic enforcement of access control policies among dynamic coalitions. In: Proceedings of the 12th ACM symposium on Access control models and technologies (SACMAT'07), Sophia Antipolis, France, ACM Press (2007) 235–244

50. Pan, C.C., Mitra, P., Liu, P.: Semantic access control for information interoperation. In: Proceedings of the 11th ACM symposium on Access control models and technologies (SACMAT'06), New York, NY, USA, ACM Press (2006) 237–246
51. Neumann, G., Strembeck, M.: Design and implementation of a flexible RBAC-service in an object-oriented scripting language. In: Proceedings of the 8th ACM conference on Computer and Communications Security (CCS'01), ACM (2001) 58–67
52. Di, W., Jian, L., Yabo, D., Miaoliang, Z.: Using semantic web technologies to specify constraints of RBAC. In: Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05), IEEE Computer Society (2005) 543–545
53. Heilili, N., Chen, Y., Zhao, C., Luo, Z., Lin, Z.: An OWL-based approach for RBAC with negative authorization. In: Proceedings of the 1st International Conference on Knowledge Science, Engineering and Management (KSEM'06), Springer Berlin/Heidelberg (2006) 164–175
54. Wang, L., Wijesekera, D., Jajodia, S.: A logic-based framework for attribute based access control. In: Proceedings of the ACM workshop on Formal methods in security engineering (FMSE '04), ACM (2004) 45–55
55. Park, J., Sandhu, R.: The $UCON_{ABC}$ usage control model. ACM Transactions on Information and System Security **7**(1) (2004) 128–174
56. Kagal, L., Finin, T., Joshi, A.: A policy language for a pervasive computing environment. In: Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY '03), IEEE Computer Society (2003) 63
57. Uszok, A., Bradshaw, J.M., Johnson, M., Jeffers, R., Tate, A., Dalton, J., Aitken, S.: KAoS policy management for semantic web services. IEEE Intelligent Systems **19**(4) (2004) 32–41
58. Squicciarini, A.C., Bertino, E., Ferrari, E., Ray, I.: Achieving privacy in trust negotiations with an ontology-based approach. IEEE Trans. Dependable Secur. Comput. **3**(1) (2006) 13–30
59. Duma, C., Herzog, A., Shahmehri, N.: Privacy in the semantic web: What policy languages have to offer. In: Proceedings of the 8th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY '07), IEEE Computer Society (2007) 109–118
60. Strader, T.J., Lin, F.R., Shaw, M.J.: Information infrastructure for electronic virtual organization management. Decis. Support Syst. **23**(1) (1998) 75–94
61. Camarinha-Matos, L.M., Afsarmanesh, H.: A roadmap for strategic research on virtual organizations. In: Processes and Foundations for Virtual Organizations, IFIP TC5/WG5.5 4th Working Conference on Virtual Enterprises (PRO-VE'03), Kluwer (2003) 33–46
62. Camarinha-Matos, L.M., Afsarmanesh, H.: A comprehensive modeling framework for collaborative networked organizations. Journal of Intelligent Manufacturing **18**(5) (October 2007) 529–542
63. Cardoso, H.L., Oliveira, E.C.: Virtual enterprise normative framework within electronic institutions. In: 5th International Workshop on Engineering Societies in the Agents World. Volume 3451 of LNCS., Springer (2004) 14–32
64. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC) **4**(3) (2001) 224–274

A Example Scenario of Interoperable Semantic Access Control

The appendix describes in details a coalition access control process based on semantic interoperability of credentials. Figure 8 shows a summary of the access control process and relevant information at the different coalition phases. There are three symbolic partners that register to share resources via a coalition platform. Phase I shows partners' semantic access policies after being registered to the platform, and being transformed to a logic program format. This phase also shows partners' defined relations of their semantic contexts.

For example, Partner B has registered resources res_{b1} and res_{b2}. Partner B's access policy specifies that access to resources res_{b1} under action act_{b1} is granted if a credential (its value represented by) c_{b1} with semantics o_{b1} and a credential value c_{b2} with semantics o_{b2} are active in the system. Analogously, access to res_{b2} under action act_{b2} is granted if a credential c_{b3} with semantics o_{b3} is given. Partner B defines also a constraint rule on the presence of credentials c_{b2} with semantics o_{b2} and c_{b3} with o_{b3}.

A coalition formation request triggers a coalition formation process that selects a coalition instance with Partner A, Partner B and Partner C. As a result, a semantic assignment policy $\mathcal{P}_\mathcal{C}$ (of Definition

4) and a semantic interoperability policy $\mathcal{P}_{\mathcal{SI}}$ (of Definition 5) are generated according to the selected partners. Phase II shows the resulting policies.

Phase III shows a snapshot of the access control process (of Figure 4) during the operation of $\mathcal{X}$. A resource res_{b1} of partner B is to be executed with an action act_{b1} on behalf of a client. The client has already presented credentials c_{a1} and c_{c1} in previous steps (or collected them in past interactions).

Step 1 of the access control process assigns semantics to the input set of credentials, that is, the credential c_{a1} is recognized with semantics in partner A's settings – $\mathsf{sem\_cred}\,(c_{a1}, o_{a1})$, and credential c_{c1} is recognized with a semantic context in partner C's settings – $\mathsf{sem\_cred}\,(c_{c1}, o_{c1})$. Step 1' generates $\hat{\mathcal{C}}_{\mathcal{SA}}$ by prefixing all semantic credentials computed in step 1.

Step 2 computes a set of semantically equivalent credentials $\mathcal{C}_{\mathcal{SE}}$ according to $\mathcal{P}_{\mathcal{SI}}$ and $\hat{\mathcal{C}}_{\mathcal{SA}}$. Essentially, partner B's credential c_{b2} is derived as equivalent to c_{a1} under the semantic context o_{b2} via the semantic relation $\mathsf{subClassOf}\,(o_{a1}, o_{b2})$. Similarly, partner B's credential c_{b1} is recognized as equivalent to c_{c1} under the semantic context o_{b1} via the semantic relation $\mathsf{equivalentClass}\,(o_{b1}, o_{c1})$.

Next, step 3 and step 4 filter out from $\mathcal{C}_{\mathcal{SA}}$ and $\mathcal{C}_{\mathcal{SE}}$ those semantic credentials not relevant for the decision process of partner B. Since $\mathcal{C}_{\mathcal{SA}}$ contains no credentials relevant for partner B's settings, $\mathcal{C}_{\mathcal{SA}}$ is set up to an empty set in step 3. Step 5 checks if the input set of credentials (with semantic assignment) and their semantically equivalent credentials satisfy the requirements of the semantic access policy of partner B for granting the request. Step 6 grants access to r.

We have seen that even though the (client's) active credentials are different from those required for the current coalition resource execution, the access decision process grants the resource execution thanks to the defined semantic interoperability of credentials. The coalition operational phase can continue smoothly with the next step of its execution.

B Instantiating a Semantic Access Policy from an RBAC Specification

The model advocated in the paper provides an access control enforcement process based on policies compliant with the coalition-wide semantic policy vocabulary. As we have noted, a partner could either specify the semantic policy directly at the remote coalition platform or generate (transform) it from an existing access control specification. In this section we describe a transformation mechanism based on RBAC settings. We summarize below core RBAC model [64].

Definition 7. (RBAC)

- *U, R, OP and O are sets of users, roles, operations and objects respectively.*
- *$UA \subseteq U \times R$ is a many to many user to role assignment relation.*
- *P the set of permissions, $P \subseteq \{(op, o) \mid op \in OP \wedge o \in O\}$.*
- *$PA \subseteq P \times R$ is a many to many permission to role assignment relation.*
- *$\texttt{assigned\_users}(r) = \{u \in U \mid (u, r) \in UA\}$, the mapping of role r onto a set of users.*
- *$\texttt{assigned\_permissions}(r) = \{r \in R \mid (u, r) \in UA\}$, the mapping of role r onto a set of permissions.*
- *$\texttt{user\_roles}(u) = \{r \in R \mid (u, r) \in UA\}$.*
- *$RH \subseteq R \times R$ is a partial order on R called the role hierarchy or role dominance ($\succeq$) relation. $r_i \succeq r_j \Rightarrow \texttt{authorized\_permissions}(r_j) \subseteq \texttt{authorized\_permissions}(r_i)$ and $\texttt{authorized\_users}(r_i) \subseteq \texttt{authorized\_users}(r_j)$.*
- *$\texttt{authorized\_users}(r) = \{u \mid r' \succeq r, (u, r') \in UA\}$.*
- *$\texttt{authorized\_permissions}(r) = \{p \mid r' \succeq r, (p, r') \in PA\}$.*
- *$\texttt{authorized\_roles\_per\_permission}(p) = \{r \mid r \succeq r', (p, r') \in PA\}$.*
- *Static Sep. of Duties: $\forall (RS, n) \in SSD, \forall T \subseteq RS : |T| \geq n \Rightarrow \bigcap_{r \in T} authorized\_users(r) = \emptyset$.*

We define a minimal extension to RBAC functionalities to make the transformation process based on attributes. We refer the reader to [49] for an approach on characterizing attributes for RBAC roles.

Definition 8. (RBAC Extension)

I. Partner registration phase

Partner A:

$\mathcal{P}_{\mathcal{SA}}$: $\mathsf{grant}\,(res_{a1}, act_{a1}) \leftarrow \mathsf{sem\_cred}\,(c_{a1}, o_{a1})$.

Partner B:

$\mathcal{P}_{\mathcal{SA}}$: $\mathsf{grant}\,(res_{b1}, act_{b1}) \leftarrow \mathsf{sem\_cred}\,(c_{b1}, o_{b1}),\ \mathsf{sem\_cred}\,(c_{b2}, o_{b2})$.
$\mathsf{grant}\,(res_{b2}, act_{b2}) \leftarrow \mathsf{sem\_cred}\,(c_{b3}, o_{b3})$.
$\leftarrow \mathsf{sem\_cred}\,(c_{b2}, o_{b2}),\ \mathsf{sem\_cred}\,(c_{b3}, o_{b3})$.

Partner C:

$\mathcal{P}_{\mathcal{SA}}$: $\mathsf{grant}\,(res_{c1}, act_{c1}) \leftarrow \mathsf{sem\_cred}\,(c_{c1}, o_{c1})$.
$\mathsf{grant}\,(res_{c2}, act_{c2}) \leftarrow \mathsf{sem\_cred}\,(c_{c2}, o_{c2})$.

Semantic context relations:

$\mathcal{R}$: $\mathsf{subClassOf}\,(o_{a1}, o_{b2})$. $\mathsf{equivalentClass}\,(o_{b1}, o_{c1})$. $\mathsf{subClassOf}\,(o_{c2}, o_{b3})$.

II. Coalition formation phase $\mathcal{X} = \{Partner\ A, Partner\ B, Partner\ C\}$.

$\mathcal{P}_{\mathcal{C}}$:
$\mathsf{sem\_cred}\,(c_{a1}, o_{a1}) \leftarrow \mathsf{cred}\,(c_{a1})$.
$\mathsf{sem\_cred}\,(c_{b1}, o_{b1}) \leftarrow \mathsf{cred}\,(c_{b1})$.
$\mathsf{sem\_cred}\,(c_{b2}, o_{b2}) \leftarrow \mathsf{cred}\,(c_{b2})$.
$\mathsf{sem\_cred}\,(c_{b3}, o_{b3}) \leftarrow \mathsf{cred}\,(c_{b3})$.
$\mathsf{sem\_cred}\,(c_{c1}, o_{c1}) \leftarrow \mathsf{cred}\,(c_{c1})$.
$\mathsf{sem\_cred}\,(c_{c2}, o_{c2}) \leftarrow \mathsf{cred}\,(c_{c2})$.

$\mathcal{P}_{\mathcal{SI}}$:
$\mathsf{sem\_cred}\,(c_{a1}, o_{a1})$.
$\mathsf{sem\_cred}\,(c_{b1}, o_{b1})$.
$\mathsf{sem\_cred}\,(c_{b2}, o_{b2})$.
$\mathsf{sem\_cred}\,(c_{b3}, o_{b3})$.
$\mathsf{sem\_cred}\,(c_{c1}, o_{c1})$.
$\mathsf{sem\_cred}\,(c_{c2}, o_{c2})$.
$\mathsf{subClassOf}\,(o_{a1}, o_{b2})$.
$\mathsf{equivalentClass}\,(o_{b1}, o_{c1})$.
$\mathsf{subClassOf}\,(o_{c2}, o_{b3})$.
$\mathsf{equivalentClass}\,(O', O) \leftarrow \mathsf{equivalentClass}\,(O, O')$.
$\mathsf{disjointWith}\,(O', O) \leftarrow \mathsf{disjointWith}\,(O, O')$.
$\mathsf{disj\_sem\_cred}\,(C, O') \leftarrow \mathsf{sem\_cred}\,(C, O),\ \mathsf{disjointWith}\,(O, O')$.
$\mathsf{sem\_cred}\,(C, O') \leftarrow \mathsf{sem\_cred}\,(C, O),\ \mathsf{subClassOf}\,(O, O')$.
$\mathsf{sem\_cred}\,(C, O') \leftarrow \mathsf{sem\_cred}\,(C, O),\ \mathsf{equivalentClass}\,(O, O')$.
$\mathsf{final\_sem\_cred}\,(C, O) \leftarrow \mathsf{sem\_cred}\,(C, O),\ not\ \mathsf{disj\_sem\_cred}\,(C, O)$.
$\mathsf{equiv\_sem\_cred}\,(C', O) \leftarrow \mathsf{given\_sem\_cred}\,(C, O),\ \mathsf{final\_sem\_cred}\,(C', O),\ C \neq C'$.
$\mathsf{equiv\_sem\_cred}\,(C', O') \leftarrow \mathsf{given\_sem\_cred}\,(C, O),\ \mathsf{final\_sem\_cred}\,(C, O'),\ \mathsf{final\_sem\_cred}\,(C', O'),\ C \neq C',\ O \neq O'$.

III. Coalition operation phase

`SemanticAccessControl`($r = \mathsf{grant}\,(res_{b1}, act_{b1})$, $\mathcal{C}_{\mathcal{A}}$=$\{\mathsf{cred}\,(c_{a1}),\ \mathsf{cred}\,(c_{c1})\}$)

1. $\mathcal{C}_{\mathcal{SA}} = \{\mathsf{sem\_cred}\,(c_{a1}, o_{a1}), \mathsf{sem\_cred}\,(c_{c1}, o_{c1})\}$;
1'. $\hat{\mathcal{C}}_{\mathcal{SA}} = \{\mathsf{given\_sem\_cred}\,(c_{a1}, o_{a1}), \mathsf{given\_sem\_cred}\,(c_{c1}, o_{c1})\}$;
2. $\mathcal{C}_{\mathcal{SE}} = \{\mathsf{sem\_cred}\,(c_{b2}, o_{b2}), \mathsf{sem\_cred}\,(c_{b1}, o_{b1})\}$;
3. $\mathcal{C}_{\mathcal{SA}} = \emptyset$;
4. $\mathcal{C}_{\mathcal{SE}} = \{\mathsf{sem\_cred}\,(c_{b2}, o_{b2}), \mathsf{sem\_cred}\,(c_{b1}, o_{b1})\}$;
5. $\mathcal{P}_{\mathcal{SA}} \cup \mathcal{C}_{\mathcal{SA}} \cup \mathcal{C}_{\mathcal{SE}} \models r$ and $\mathcal{P}_{\mathcal{SA}} \cup \mathcal{C}_{\mathcal{SA}} \cup \mathcal{C}_{\mathcal{SE}} \not\models \bot$
6. *grant* r.

Fig. 8. Example scenario of interoperable semantic access control process

Algorithm 1 RBAC to semantic access policy

```
1: generate_resourcepolicy(p){
2: new ResourcePolicy, PolicyRequirements;
3: for any r ∈ authorized_roles_per_permission(p) do
4:    new CredentialSet;
5:    for any a ∈ assigned_attributes_per_role(r) do
6:       ap = assigned_attribute_provider_per_attribute(a);
7:       CredentialSet.add( generate_credential(a, ap) );
8:    end for
9:    PolicyRequirements.add( CredentialSet );
10: end for
11: ResourcePolicy.add( generate_protectedresource(p) );
12: ResourcePolicy.add( PolicyRequirements );
13: return ResourcePolicy;
14: }
15: generate_accesspolicy(⟨p_1, ..., p_n⟩){
16: new AccessPolicy, AccessPolicyConstraints;
17: new AccessPolicyBody, AllAuthorizedRoles;
18: for any p ∈ ⟨p_1, ..., p_n⟩ do
19:    AccessPolicyBody.add( generate_resourcepolicy(p) );
20:    AllAuthorizedRoles.add( authorized_roles_per_permission(p));
21: end for
22: for any (rs, n) ∈ ssd do
23:    for any s ⊆ rs and |s| ≥ n do
24:       for any ssd ∈ all_authorized_roles_per_ssd_set(s) do
25:          if ssd ⊆ AllAuthorizedRoles then
26:             new CredentialSet;
27:             for any r ∈ ssd do
28:                for any a ∈ assigned_attributes_per_role(r) do
29:                   ap = assigned_attribute_provider_per_attribute(a);
30:                   CredentialSet.add( generate_credential(a, ap) );
31:                end for
32:             end for
33:             AccessPolicyConstraints.add( CredentialSet );
34:          end if
35:       end for
36:    end for
37: end for
38: AccessPolicy.add( AccessPolicyBody );
39: AccessPolicy.add( AccessPolicyConstraints );
40: return AccessPolicy;
41: }
42: RBAC_TO_SEMANTIC_POLICY(⟨p_1, ..., p_n⟩){
43: AccessPolicy = generate_accesspolicy(⟨p_1, ..., p_n⟩);
44: SemAccessPolicy = assign_sem_contexts(AccessPolicy);
45: InstSemPolicy = instantiate_policy(SemAccessPolicy);
46: return InstSemPolicy;
47: }
```

- *A and AP are sets of attributes and attribute providers respectively.*
- *$APA \subseteq A \times AP$ many-to-one attribute to attribute provider assignment relation.*
- ***assigned_attribute_provider_per_attribute***$(a) = \{ap \in AP \mid (a, ap) \in APA\}$.
- ***assigned_attributes_per_role***$(r) = \{a \mid a \in A\}$ *a function that determines a set of attributes characterizing a role r. If no attributes assigned the function returns the role itself as the only attribute.*
- ***all_authorized_roles_per_ssd_set***$(\langle r_1, \ldots, r_n \rangle) = \{\langle r_1^{'}, \ldots, r_n^{'} \rangle \mid r_i^{'} \succeq r_i, 1 \leq i \leq n\}$ *a function that returns all n-arity tuples with roles dominating their respective position-roles in the input tuple.*

Many-to-one attribute to attribute provider assignment states that an attribute is provided by a single authority. Multiple authorities responsible for an attribute could also be supported by extending the coalition policy vocabulary and the transformation mechanism.

Algorithm 1 shows a semantic access policy generation for a set of resources (objects and operations) a partner wishes to share in a coalition. We show only SSD constraints in the transformation process. One can approach dynamic separation of duties (DSD) by extending the AccessPolicyContraints class of the coalition vocabulary with the notion of DSD, and then perform analogous transformation to that of SSD.

Function `generate_accesspolicy`($\langle p_1, \ldots, p_n \rangle$) generates an access policy tree in accordance to the coalition policy vocabulary. For any permission p_i the function generates a ResourcePolicy element and then updates a set of all authorized roles per all permissions. The last set is used for AccessPolicyConstraints generation.

Function `generate_resourcepolicy`(p) takes as argument a permission p and for any role authorized for that permission it generates a CredentialSet element. Then, for any attribute characterizing an authorized role, the function generates a Credential node element that is added to the CredentialSet. In this way, a CredentialSet element encapsulates all necessary attributes for a role authorized for the permission p.

Lines 22–37 of function `generate_accesspolicy` define the steps for transforming SSD constraints to a semantic description. Lines 22–23 define SSD role set configurations according to the RBAC SSD definition. Next, for any SSD role set (lines 24–25) we generate all authorized role sets of the SSD set so that out of them we select only those configurations which have relevant scope to the protected permissions, i.e. which are subset or equal of all authorized roles of the protected permissions.

For any SSD set with a relevant scope, we generate a CredentialSet element (lines 26–33) that encapsulates the attributes characterizing all roles in the set. Essentially, lines 27–32 generate a Credential node for each attribute of a role in the SSD set.

The end-function `generate_credential`(*attribute*, *provider*, *holder*, *semcontext*) generates a node of the access policy tree that encapsulates a Credential class and its properties.

Function `generate_protectedresource`(*permission*) generates a node of the access policy tree that encapsulates ProtectedResource class with its properties.

Function `assign_sem_contexts`(*AccessPolicy*) provides a GUI to facilitate the administrator when assigning semantic contexts to credentials. Since context generation and assignment may depend on multiple factors, for example on coalition type, partners participation, resources and actions etc, we abstract assignment details by leaving the security administrator to define and assign them with the aid of semi-automatic tools. However, if semantic context assignment depends only on resources and actions, and attributes (roles) used, one can automate the assignment process by defining an appropriate function that assigns contexts to attributes.

Function `instantiate_policy`(*SemAccessPolicy*) generates an OWL/RDF instance of the coalition policy vocabulary. It instantiates all classes of nested elements corresponding to the description of the vocabulary, for example, a Credential class will be instantiated by first instantiating Holder, Attribute, Provider and SemContext classes.