JUET Research Journal of Science & Technology
6 (Special Issue) 2020, 11-21
http://www.publishingindia.com/juet

An Approach to Design Reference Ontology for Access Control Models

Ravindra Kumar Singh[1*] and Reena Singh[2]

[1]Computer Science and Engineering, Jaypee University of Engineering and Technology, Guna, Madhya Pradesh, India. Email: ravindra.singh@juet.ac.in
[2]Information and Communication Technology, Manipal Institute of Technology, Manipal, Karnataka, India.
*Corresponding Author

Abstract: Many access control models (ACM) have been proposed in the literature pertaining to specific requirements of different organisations and scenarios. Few approaches were made to provide better understandability of different aspects of ACM, like constraints, with the help of ontology. A reference ontology is needed to provide common semantics to understand and compare different ACMs of different systems and organisations which collaborate to work together like in Computer Supported Collaborative Work (CSCW). In this paper, we identify the common concepts across different ACMs and propose a generic ontology based on these concepts. Specific ontologies of different ACMs can be derived from this reference ontology. We demonstrate its applicability in some examples and make suggestions for its adaptability to suit more generic applications and scenarios.

Keywords: Access control model, Meta-model, Ontology.

I. Introduction

Every organisation's security is dictated by the access control policy and defined in terms of the access control model. Access control becomes a necessity in the networked systems where users from one or different organisations request access to the resources on the network. Access requests are made by either persons or services on behalf of the person. The resource may be file, directory, database, or a service. The basic entities in access control models are user, action, and object. An access control controls who can perform what actions on which object. A plethora of access control models have been proposed over the past decade for different application areas. What started as Lampson's access control matrix [1] has developed over the ages to make Role based access control (RBAC), Attribute based access control (ABAC), Team based access control (TBAC), Content based access control (CBAC), Purpose based access control (PBAC) and so on so forth. The difference between these different access control models is the abstraction they provide to the basic concepts (user, action, and object). For instance, instead of defining access for individual users, users are assigned to 'roles' based on the job functions or responsibility in an organisation and access is defined in terms of roles, making it easy to administer large number of users. Although the needs and representation of access control differ, these models have many aspects in common which could be utilised to make a meta-model such that different access control models could be formulated from it. This question was raised by Ferraiolo in [2].

A similar argument was given by Landin [3] in favour of identifying a set of programming language primitives and instantiating different programming languages from it, by selecting different subsets. Some research work has been done to use ontology for access control models like [4].

An ontology represents the vocabulary of concepts in a domain and the relationship among those concepts. Ontologies represent agreed domain semantics and are independent of specific applications. This makes them reusable by different applications. By sharing an ontology, autonomous and distributed applications can meaningfully communicate to exchange data independently of their internal technologies. Jarrar *et al.* in [5] discuss on the usability versus reusability of ontologies and propose to formulate ontology by separating intended models of a vocabulary at the domain level from the usability of this vocabulary according to certain application/ usability perspectives and specify the legal models (a subset of the intended models) of the application(s) interest.

In this paper, we try to attempt to frame a generic ontology from which ontology of many common access control models can be instantiated. We also discuss the various advantagesof this approach and present a mechanism for inter-conversion between different ontologies formed as such.

The rest of the paper is organised as follows: Section II discusses the related work in this area. Section III surveys through the existing ACMs and identifies common concepts across them. Section IV defines the reference ontology from the identified concepts, Section V derives specific ACM ontologies from the reference ontology and discusses the benefits of this approach and finally we conclude in Section VI.

II. Related Work

There have been very few proposals focusing on different aspects of design of ontology for access control models facilitating interoperability and understanding. Tsai and Shao in [6] proposed a RBAC model using a role ontology for Multi-Tenancy Architecture (MTA) in clouds, defines an ontology tree and provide algorithms to compare the similarity of different ontology trees. Finin *et al.* in [7] proposed the ways to support the Role-Based Access Control (RBAC) model in Web Ontology Language (OWL) and discussed how the OWL constructions can be extended to model attribute-based RBAC or more generally Attribute-based access control (ABAC). Di *et al.* in [8] discusses the use of a Semantic web technology, namely, OWL to specify RBAC constraints. OWL is a semantic markup language, which provides formalised knowledge expression and more flexibility, sharing a great deal of common semantics about expressing access control constraints.

III. Meta-Model of Access Control Models

Access control models, proposed in the literature, can be identified with some common concepts. This section introduces the motivation for a meta model for ACMs and discusses different ACMs in terms of the common concepts.

A. Why an Access Control Meta Model?

Access control models are used in a variety of computers and systems to restrict access to, or actions performed on data resources. Each organisation defines and enforces access control on all the networked resources to ensure authorised use by different users, as per the security policies outlined for the organisation. Interacting across diverse, heterogeneous systems are an indispensable part of access of resources on any network, be it within an organisation or between organisations or the Internet. Diverse tools supporting computer-supported collaborative work (CSCW) facilitates easy collaboration among different organisations. All these diverse systems and organisations have their own access control models (ACM). It is not feasible to insist on one ACM for all. Understanding, reasoning, or analysing these systems requires something common across these ACMs, i.e. a meta-model. This has led to the question asked by Ferraiolo and Atluri in [2], of whether it is possible, and desirable, to have a meta-model, capturing all the common semantics of different access control systems. In [4], it is argued that having such a unified framework for access policies can greatly reduce the burden of policy administration across different access control systems, as the general access control requirements would be represented in the meta model. Domain-specific requirements would be defined by specialising the general access control axioms in the meta model. As a meta-model represents an abstract syntax, and the instances of a meta model represents concrete syntaxes, having an access control meta-model opens for the possibility of many domain-specific access control syntaxes.

B. Brief Discussion of ACMs

Different access control systems are all designed to grant or deny access to resources, according to some sort of policy. The access control systems, even if they are used in separate contexts, are built on common principles, and operate with common concepts such as users, actions, and resources. Some access control systems may operate with supplemental concepts, such as user groups, action types, and resource classes. However, it is important to note, that the user is the same, although represented in many ways throughout the course of using the different access control mechanisms, since the systems are implemented independently of each other. In this section we identify these common concepts across some ACMs in the literature.

- *Mandatory Access Control (MAC) Model [9]:* Users are given security clearance and all objects are given security labels. Based on these two, Read and Write actions are allowed.
- *Discretionary Access Control Model (DAC) Model [1]:* Owners of the objects can give read, write and execute permission to different users and user groups. Every object has a Access control list (ACL) where name of user, group and mode (read, write, execute) is mentioned. Access rights can be passed from one user to another.
- *Role-based Access Control (RBAC) Model [10, 11]:* Users are assigned to roles which are job functions a user takes up in an organisation. These roles are given permission on objects or a group of objects (object type). The exact detail of permission is left to implementation.
- *RBAC Variant 1 [12]:* Proposed a variant of RBAC, where access is allowed to subset of an object based on parameter (attribute) evaluation (content-based), say access to a particular table row based on the value of table field. Suitable for database scenario.
- *Attribute-based Access Control (ABAC) Model [13, 14]:* Access is based on the attributes of the user, object and environment. A rule is evaluated based on the value of attributes and access decision is made at the run-time.
- *Team-based Access Control (TMAC) Model [15]:* It is based on RBAC model. Access permission is given to the roles based on 'team membership'. Exact details of permission is not specified.
- *Team-and-Role BAC Model (TRBAC) [16]:* Proposes categories of user, object and action to take advantage of fine- and coarse-grained access control namely, user as Individual, Role and Group (Team), object as Individual (Instance), Attribute (Type), Group (collection of same or different types) and action as Individual (Query, Update, Execute, and Assign).
- *Generalised RBAC (GRBAC) Model [17]:* This model is an extension of RBAC by incorporating object roles and environment roles along with the user roles as in RBAC. Supports content-based access using object roles.

- *Temporal RBAC (TRBAC) Model [18]:* This model supports periodic role enabling and disabling, and temporal dependencies among such actions. Such dependencies expressed by means of role triggers. This model uses the concept of periodic expressions and calendars.
- *Context based TMAC (C-TMAC) Model [19]:* Proposed by this model provides a framework to integrate RBAC, TMAC and Context (location, time, etc.). Proposes model, does not prove properties.
- *Content based Access Control (CoBAC) Model [20]:* Proposed fine-grained authorisation of user and user-group for video content access in a video database with annotations. Access is grouped as grant or deny.
- *Organization based (OrBAC) Model [21]:* This ACM observes the importance of 'organisation' in access control model along with roles and attributes and defines permissions based on these.
- *Fine-Grained ACM (FAC) Model [22]:* This model addresses access control of web services. Users and web services have attributes, and full or partial authorisation is given based on the result of policy evaluation pertaining to the service and user attributes and the request is accepted, rejected or negotiated.
- *Generalized Temporal RBAC (GTRBAC) Model [23]:* This model is an extension of TRBAC. It allows expressing periodic as well as duration constraints on roles, user-role assignments, and role-permission assignments. It supports constraints on the maximum active duration allowed to a user and the maximum number of activations of a role by a single user within a particular interval of time.
- *Purpose based Access Control (PBAC) Model [24]:* Objects are given 'purpose', say, marketing, sales, and access permission is assigned to roles based on the purpose for which they want to use the objects.
- *Activity-based Access Control (ActBAC) Model [25]:* Access is given to user based on his attributes and the activity she wants to perform with the object or object group. Role is considered as a user attribute in this model.
- *Role and Attribute based Access Control (RBAC-A) Model [26]:* This favours the access control based on roles and attributes, so that static permissions are assigned to roles and restricted by dynamic attributes (like location and time), making RBAC suitable for distributed applications.
- *ACM for Mobile Physical Objects (MoACM) [27]:* Propose Access control for data collected about mobile physical object based on the object's trajectory-based visibility policies. Subjects are identified with roles and attributes and objects or object groups with attributes and access is either allowed or denied.
- *Authorization based Access Control (ZBAC) [28]:* Proposes an access control model that uses authorisations (of the user-domain) presented with the request which is used by the other domains' authorisation servers to make an access decision, i.e. Semi-distributed authorisation.

All these approaches to access control operate with three pillar concepts: users, objects and actions. Some of them refine the definition by operating with additional abstract concepts, to make the policy definition more detailed. Thus, the concept of a group classifies users as part of a greater unity, for which certain policy rules can be specified. The concept of roles adds a layer between users and permissions. Other access control models add other concepts, such as temporal restrictions, purpose, activity, etc. Moreover, the concepts context and attributes are used interchangeably in a few ACMs. Role is a separate concept describing the job functions and responsibilities to which permissions are assigned while in some models, it is used as a user attribute evaluated to determine access. Communicating common semantics needs a non-conflicting, standard vocabulary of the base concepts which are shared and agree-upon by all the systems or organisations involved and which can be extended to make additional concepts in a restrictive way.

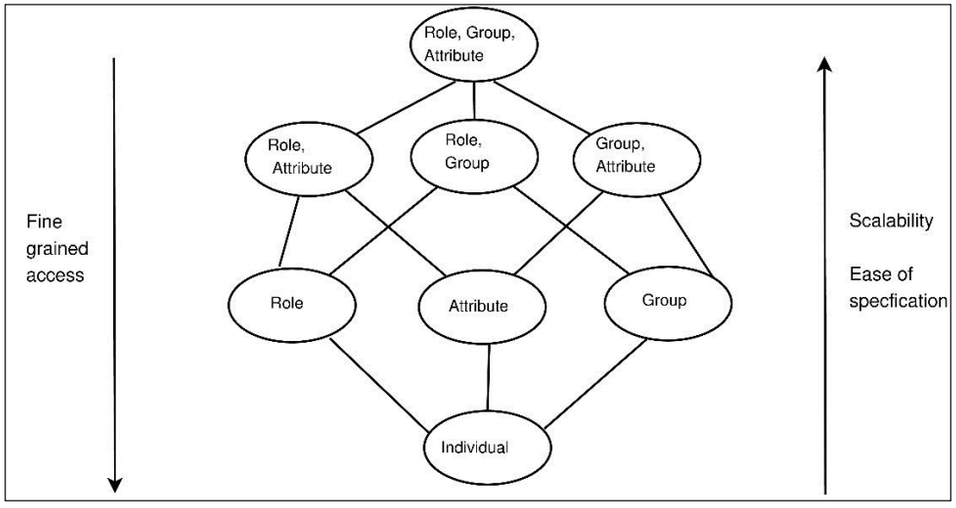

Fig. 1: Abstraction of User and Object Concepts

Table I: Interpretation of the Concepts in Different Access Control Models

ACM	User	Object	Action	Environment	Remarks
MAC	Attribute (Security Clearance)	Individual (Security labels)	Individual (Read, Write)	No	Information flow control in hierarchy
DAC	Individual, Group	Individual	Individual	No	Owner decides access, transfer of access rights
RBAC	Role	Individual	Group (Permission)	No	Constraints: SoD, PoLP
RBAC'	Role, Attribute	Individual, Attribute	Group (Permission)	No	Concept of parametrised permission
TMAC	Role, (Team) Group	Group (Type)	Group (Permission)	No	-
TRBAC	Individual, Group, Role	Individual, Attribute, Group	Individual	No	-
ABAC	Attribute	Attribute	Group (Permission)	Attribute	Rule evaluation at run-time
TRBAC'	Role	Individual	Group (Permission)	No	-
GRBAC	Role, Attribute	Role	Group (Permission)	Role	-
OrBAC	Role, Group	Individual, Group	Group (PROP)	Group	Constraints
FAC	Attribute	Attribute	Group	No	-
GTRBAC	Individual, Group, Role	Individual, Attribute, Group	Group (Permission)	No	-
CoBAC	Individual, Group	Attribute, Group (Content)	Group (Allow, Deny)	No	-
PBAC	Role, Attributes	Group (Type), Attribute	Group (Permission)	Attribute	-
ActBAC	Individual, Attributes	Group	Group (Permission)	No	
RBAC-A	Role, Attribute	Attribute	Group (Permission)	Attribute	-
MoAC	Role, Attribute	Individual, Group, Attribute	Group (Allow, Deny) (Allow, Deny)	Attribute (omitted from specification)	-

C. Comparison of ACMs

Fig. 1 shows the abstraction of the concepts in terms of fine-grained access, ease of specification and scalability. In this figure, we show the possible ways of abstracting users in an ACM. The finer the access allowed, the more the number of individual access rights be specified for the users in an organisation. For further discussions, we use only one level of abstraction (role, group, and attributes) and rest is abstracted as 'Composite' concept, for simplicity. As observed from the Table I, many of the same concepts are used in different models (using different terminology). For instance, team, purpose, activity, all these are concepts to identify and group users of similar interests and objectives to assign access rights to the group which in turn determines members' access rights. Some concepts are even used for describing different concepts in different models, for instance, object roles and object attributes are interpreted as different concepts even though they take same values as object type. On a similar note, context is used to refer to attributes and roles in few other ACMs. Therefore, a common lexicon (and ontology) needs to be defined.

IV. Proposed Ontology

The necessity of a reference ontology can be argued from the fact that every organisation needs to adopt or develop an access control model to control access to the networked resources. It would be helpful if rather than building it from the scratch, a meta model can be referred and adapted based on the specific needs. By having a reference ontology, different organisations involved in a collaboration can compare their specific ontologies of the employed access control model in terms of modularity of access control (fine-grained or coarse-grained) and the middle-layer be fine-tuned (by the administrator) to Fig. 4: Abstraction of the Concept: Action map the difference of access modularity, which is otherwise very difficult to do in the absence of a reference ontology.

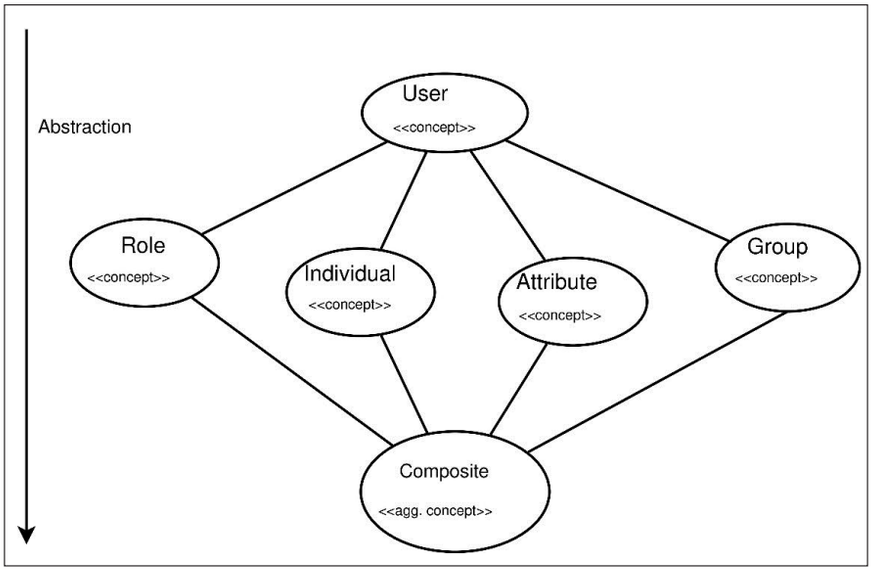

Fig. 2: Abstraction of the Concept: User

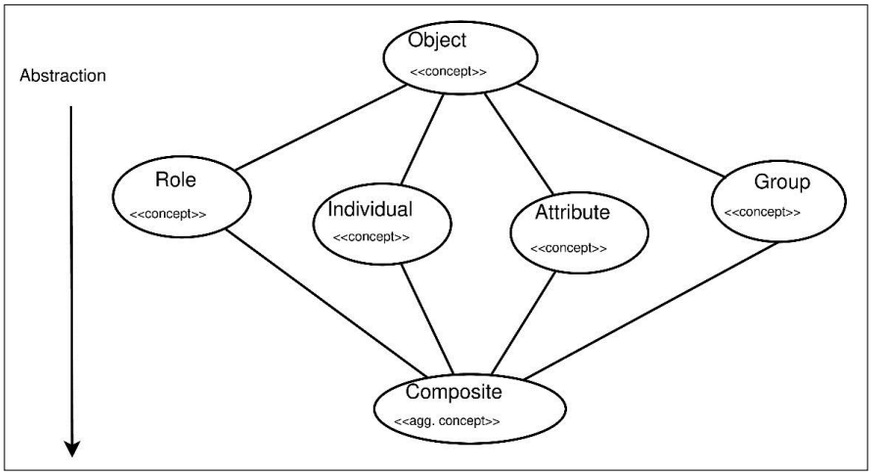

Fig. 3: Abstraction of the Concept: Object

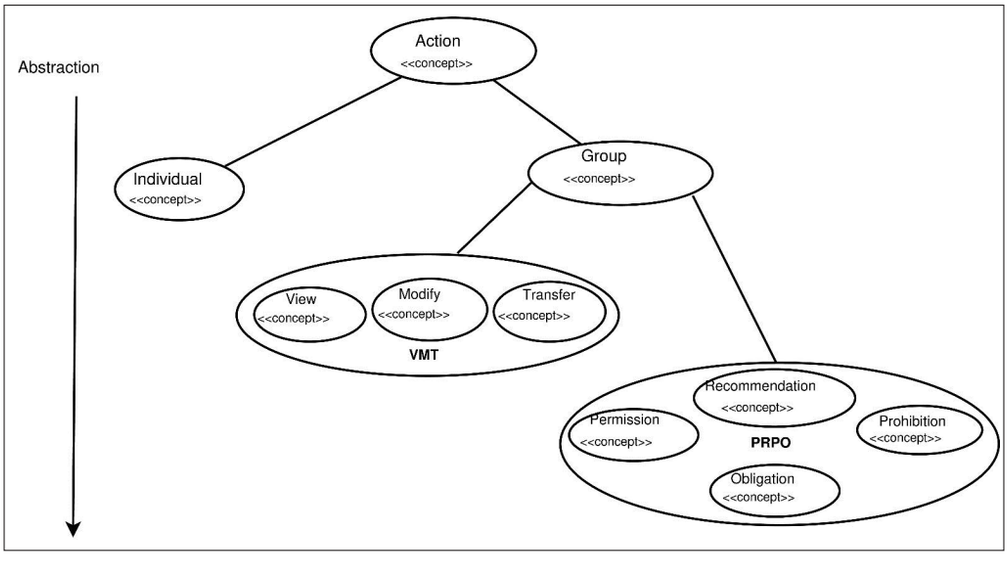

Fig. 4: Abstraction of the Concept: Action

We propose a reference ontology for access control model. Our approach of designing the reference ontology uses the method of ontology engineering proposed by Jarrar *et al.* in [5] named as DOGMA (Developing Ontology-Grounded Methods and Applications) framework. The reference ontology design has two parts- One, we define the concept of the concepts involved in the access control models in detail, and two, we establish the relationship between these concepts.

A. Description of Concepts of Ontology

In this section, we describe the semantics and interpretations of the concepts which are abstracted in specific access control model ontologies. These concepts form the basis for developing the reference ontology:

- User: The concept representing the subject of access.
- Object: The concept representing an object of access.
- Action: The concept representing the mode of access on object by the user.
- Environment: The concept representing the effect of environment on the access control model.
- Administration: The concept representing the management of access control policies.
- Authentication: The concept representing the access control enforcement.

Along with these abstract concepts, there are three other components of any ACM and hence AC ontology: Rules, Constraints and ACPolicy. ACPolicy is a part of security policy of any organisation and any ACPolicy is represented in terms of ACM. ACPolicy defines rules and constraints.

Fig. 2 represents the abstractions of the concept 'User' in the ACM. Individual is the abstraction of a person or service by name, id or other identifiers. Role is the responsibility or job function of a user in any organisation. Users are assigned to Roles and roles decide the access given to any user. Attributes are the properties (age, department, etc.), certificates or credentials possessed by the user, which are decisive factors in authorisation. User may be a member of a Group and hence get access rights of the group. The concept 'Group' may be interpreted as team, department, division, organisation, community, section, country, etc. in specific ACMs. Another abstraction can be a combination of these concepts, termed as Composite abstraction, for example, role and attributes together may determine the access.

Fig. 3 represents the abstractions of the concept 'Object' in the ACM. Individual is the abstraction of a file or service by name, id or other identifiers. Objects may be grouped together based on some conditions, like author, content, etc. Access rights of the object is the same as that of the group it is a member of. Access can be based on the object's attributes, say, type or size; hence attributes become another abstraction of the concept Object. Composite abstraction combines one or more of these concepts to together determine the access.

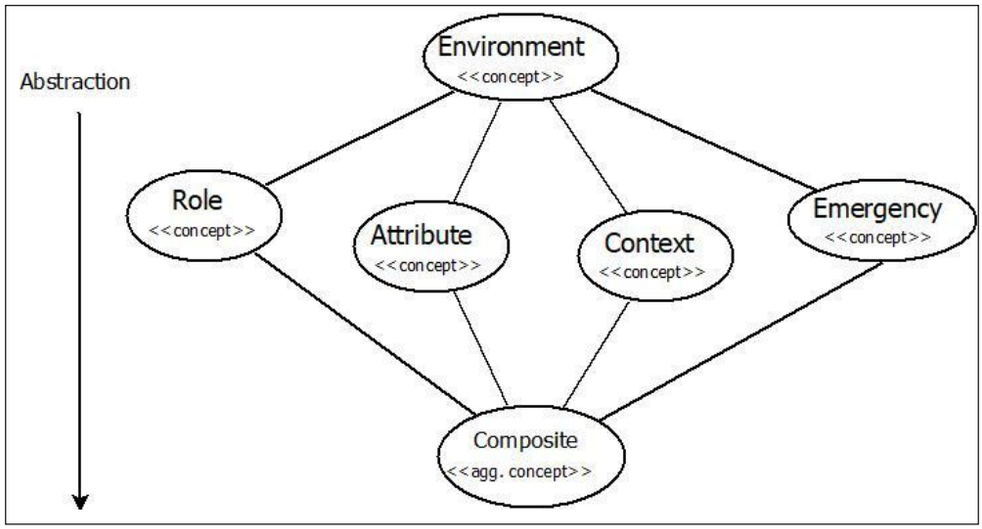

Fig. 5: Abstraction of the Concept: Administration

Fig. 4 represents the abstractions of the concept 'Action' in the ACM. Action can be specified as Individual- Read, Write, Execute, Append, etc., or as Groups. One abstraction of group is View, Modify and Transfer grouping of actions. View states that the object can be only viewed, Modify implies that the object can be viewed and edited while transfer states that the object can be passed from one user to another without viewing or modifying. Another group abstraction is Permission, Obligation, Recommendation and Prohibition.

Fig. 5 represents the abstractions of the concept 'Environment' in the ACM. Anything other than user and object form the environment. No effect abstracts the fact that this concept does not determine access. Role and Attributes (location, time) are the other abstractions. Context is another abstraction and may be interpreted as marketing, report generation, patient admittance, etc. Emergency is another abstraction which determines access in case an emergency occurs. These concepts can combine and determine access ad composite abstraction.

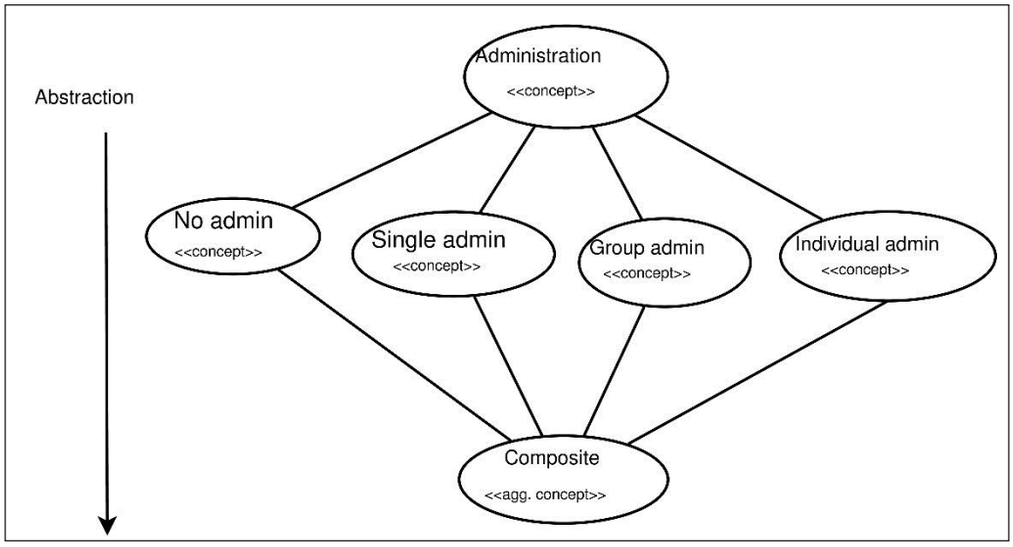

Fig. 6: Refinement of the Concept: Administration

In Fig. 6 the concept 'Administration' refers to management of the access control policies. There may be no admin of the policies, or a single (security administrator) responsible for administration (as in RBAC [Sandhu *et al.*, 1996]). A group of administrators may be responsible for management or the users may manage the policies (corresponding to them).

In Fig. 7 the concept 'Authentication' refers to enforcement of the access control policies. There may be no authorisation and free access to all. Or there may be one central server responsible for authorisation (as in RBAC). Authorisation may be distributed among different servers such that they authorise a subset of users i.e. semi-distributed (as in ZBAC [28]). Further, each object may have an authorisation wrapper (having access policy corresponding to that object) with which users are authorised i.e. fully distributed. Now, we develop the ontology base of the concepts mentioned above (user, object, action, and environment) according to the DOGMA approach [5]. Tables II to V show the lexons of the concepts of ACM abstractions, used in instantiating specific ACMs from the reference ontology.

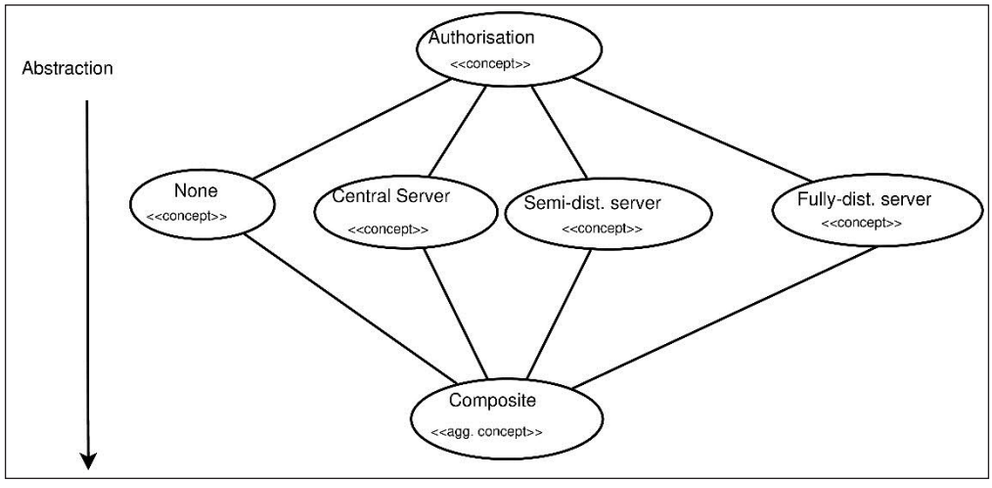

Fig. 7: Refinement of the Concept: Authorization

B. Reference Ontology

We now propose the reference ontology based on the concepts discussed in Section A of IV. Table VI describe the lexons representing ontology base for the Reference Ontology in Fig. 8.

Following rules should be referred to instantiate reference ontology into specific ontologies for different ACMs.

- No Concept should be deleted.
- The mandatory concepts- User, Action and Object have to be specified in accordance with the lexons.

- Environment, an optional concept, can be not instantiated, in which case it is implied that the environment has no effect in the ACM.
- In case a new concept needs to be introduced, a relation should be defined for the new concept and an existing concept.

Derive ontology from the meta model above. Having done that, specific access control ontologies can be derived from the reference ontology which is shown in the Section V.

Table II: Lexons Representing the Concept of User

Context	Term 1	Relation	InvRelation	Term 2
ACM	User	is-a	is-a	Individual
ACM	Name	defines	is-defined-by	Individual
ACM	Identity	defines	is-defined-by	Individual
ACM	User	is-assigned-to	of	Role
ACM	User	has	of	Attribute
ACM	User	belong-to	has	Group
ACM	User	invokes	is-invoked-by	Service
ACM	User	is-a	is-a	Person

Table III: Lexons Representing the Concept of Object

Context	Term 1	Relation	InvRelation	Term 2
ACM	Object	is-a	is-a	Individual
ACM	Object	has	of	Role
ACM	Object	has	of	Attributes
ACM	Object	belong-to	has	Group

Table IV: Lexons Representing the Concept of Environment

Context	Term 1	Relation	InvRelation	Term 2
ACM	Environment	has	of	Role
ACM	Environment	has	of	Attributes
ACM	Environment	has	of	Context
ACM	Environment	is	in	Emergency

V. Specific Access Control Model's Ontologies

Fig. 9 shows the ontology for the basic RBAC model [11]. Fig. 10 shows the ontology for the basic TMAC model [15]. Similarly, ontologies of other ACMs can be derived by referring to the Table I, corresponding lexons and rules specified in Section B of IV.

Fig. 9 and 10 show the ontologies of RBAC and TMAC derived from a common reference ontology. The individual ACMs could operate together (as a part of different systems interacting together) if a mapping from them to the reference ontology is created. Since the concepts of ACMs is common, known concept to the systems (though it might be aliased by the specific ACMs), various ACMs can inter-operate properly.

Table V: Lexons Representing the Concept of Action

Context	Term 1	Relation	InvRelation	Term 2
ACM	Action	is-a	is-a	Individual
ACM	Read	is-a	is-a	Individual
ACM	Write	is-a	is-a	Individual
ACM	Read-Write	is-a	is-a	Individual
ACM	Permission	implies	is-implied-by	Action
ACM	Obligation	implies	is-implied-by	Action
ACM	Prohibition	implies	is-implied-by	Action
ACM	Recommendation	implies	is-implied-by	Action

Table VI: Lexons Representing the Concept of Reference Ontology

Context	Term 1	Relation	InvRelation	Term 2
ACM	User	performs	is-performed-by	Action
ACM	Action	acts-on	is-acted-upon-by	Object
ACM	ACPolicy	specifies	is-specified-by	Condition
ACM	ACPolicy	has	of	Identifier
ACM	ACPolicy	written-by	writes	Admin
ACM	ACPolicy	has	of	Description
ACM	ACPolicy	defines	is-defined-by	Rule
ACM	ACPolicy	defines	is-defined-by	Constraints
ACM	Authorisation	authorises	is-authorised-by	User
ACM	Administration	maintains	is-maintained-by	ACPolicy
ACM	Constraint	specifies	is-specified-by	Rule
ACM	Rule	governs	is-governed-by	User
ACM	Rule	governs	is-governed-by	Object
ACM	Rule	governs	is-governed-by	Action
ACM	Constraint	specifies	is-specified-by	Environment

VI. Conclusion and Future Work

All the systems and organisations use ACM to control access of the resources present in the network. Many ACMs exist

in the literature and different systems use different ACMs to suit their needs. It is infeasible to insist on having one ACM for all which makes it important to have a reference ontology from which specific ACM ontologies can be derived. Ontology is very helpful in sharing semantics across diverse systems without including internal details. Previous approaches of using ontology in ACM addresses only limited aspects like constraints. In this work, we identified the common concepts across different ACMs and defined a reference ontology based on these concepts. We also derived specific ACM ontologies from the reference ontology and discussed the advantages of this approach.

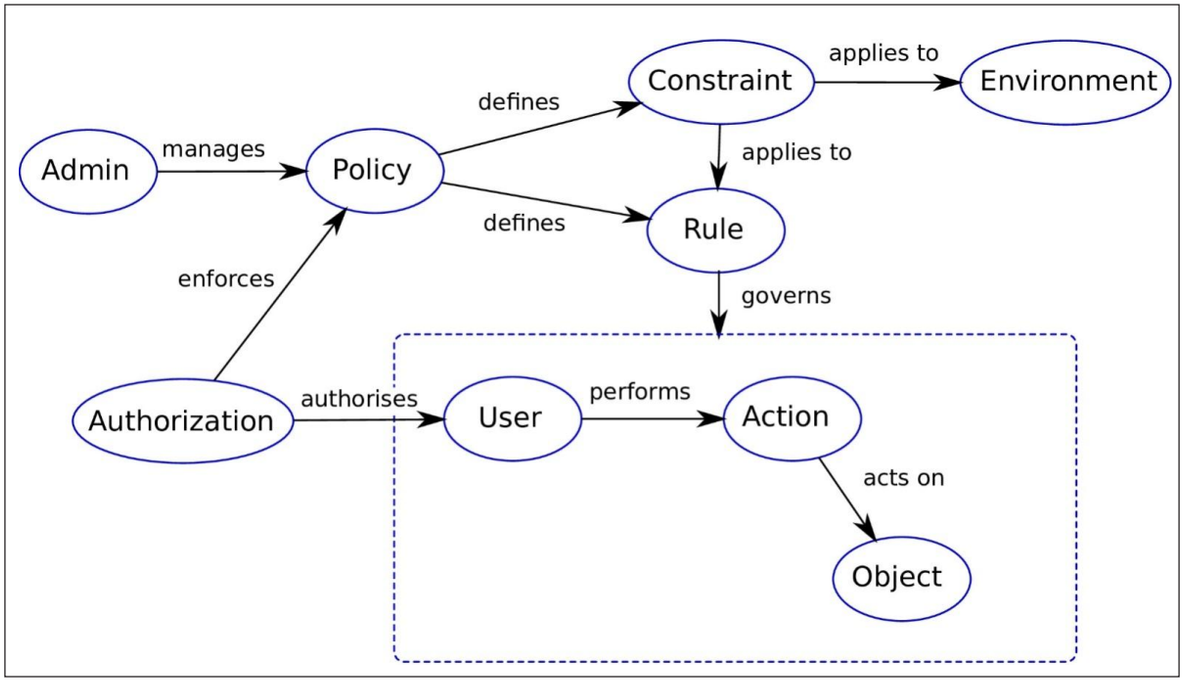

Fig. 8: Reference Ontology

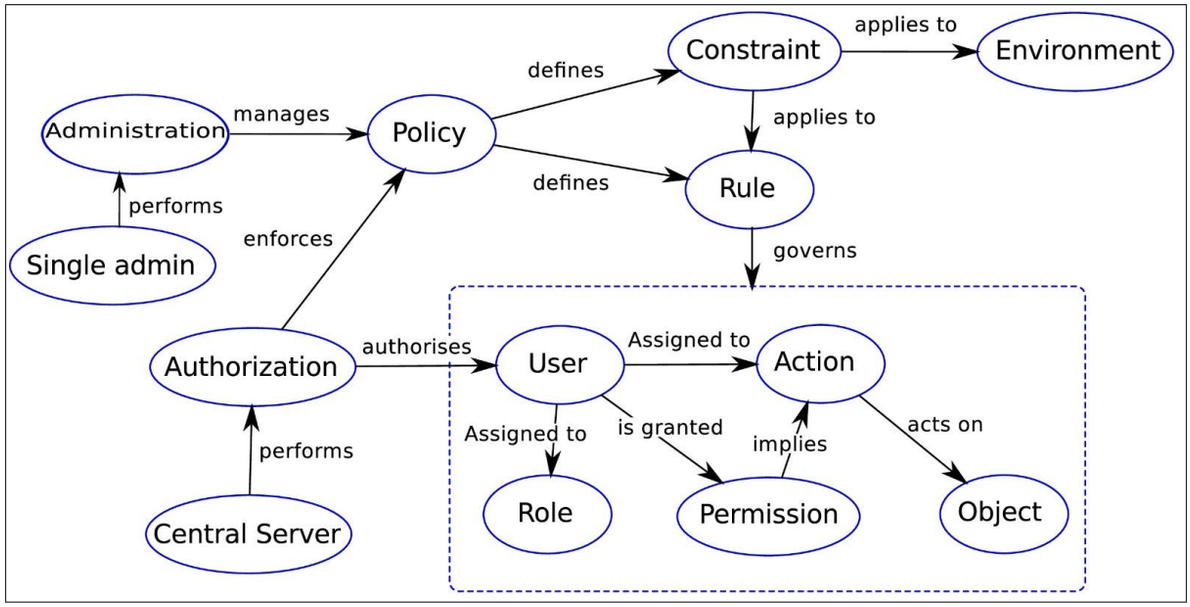

Fig. 9: RBAC Ontology

This approach of defining a reference ontology in terms of the concepts across ACMs is first of its kind in the ACM literature. As a next step, we plan to refine the reference ontology, enhance the lexons and provide precise rules for specific ACM ontology derivation (including the mapping across different ACMs). We also plan to understand and model the concepts of rules, constraints and ACPolicies across different AC enforcements.

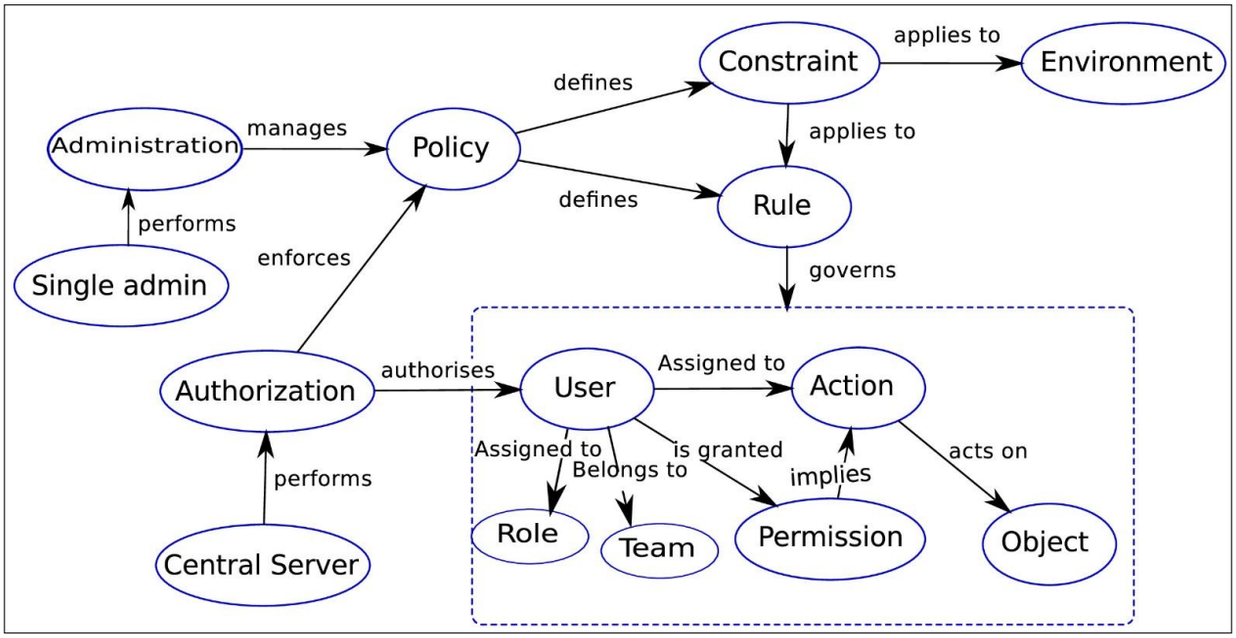

Fig. 10: TMAC Ontology

References

[1] C. McDowell, "Protection at the micromachine level," *SIGARCH Comput. Archit. News*, vol. 10, no. 1, pp. 4-8, 1982.

[2] D. Ferraiolo, and V. Atluri, "A meta model for access control: Why is it needed and is it even possible to achieve?," in *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*, Association for Computing Machinery: Estes Park, CO, USA, 2008, pp. 153-154.

[3] P. J. Landin, "The next 700 programming languages," *Commun. ACM*, vol. 9, no. 3, pp. 157-166, 1966.

[4] S. Barker, "The next 700 access control models or a unifying meta-model?," in *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies,* Association for Computing Machinery: Stresa, Italy, 2009, pp. 187-196.

[5] M. Jarrar, and R. Meersman, *Ontology Engineering – The DOGMA Approach*, in *Advances in Web Semantics I: Ontologies, Web Services and Applied Semantic Web*, T. S. Dillon, et al., Ed., Springer Berlin Heidelberg: Berlin, Heidelberg, 2009, pp. 7-34.

[6] W. Tsai, and Q. Shao, "Role-based access-control using reference ontology in clouds," in *2011 Tenth International Symposium on Autonomous Decentralized Systems*, 2011.

[7] T. Finin, et al., "ROWLBAC: Representing role based access control in OWL," in *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*, Association for Computing Machinery: Estes Park, CO, USA, 2008, pp. 73-82.

[8] D. Wu, et al., "Using semantic web technologies to specify constraints of RBAC," in *Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05)*, 2005.

[9] D. E. Bell, and L. J. L. Padula, *Secure Computer Systems: Mathematical Foundations and Model*. Mitre Corporation, 1973.

[10] D. F. Ferraiolo, J. F. Barkley, and D. R. Kuhn, "A role-based access control model and reference implementation within a corporate intranet," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 1, pp. 34-64, 1999.

[11] R. S. Sandhu, et al., "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38-47, 1996.

[12] L. Giuri, and P. Iglio, "Role templates for content-based access control," in *Proceedings of the Second ACM Workshop on Role-Based Access Control*, Association for Computing Machinery: Fairfax, Virginia, USA, 1997, pp. 153-159.

[13] P. A. Bonatti, and P. Samarati, "A uniform framework for regulating service access and information release on the web," *J. Comput. Secur.*, vol. 10, no. 3, pp. 241-271, 2002.

[14] E. Damiani, S. D. C. D. Vimercati, and P. Samarati, "New paradigms for access control in open environments," in *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology,* 2005.

[15] R. K. Thomas, "Team-based access control (TMAC): A primitive for applying role-based access controls in collaborative environments," in *Proceedings of the Second ACM Workshop on Role-Based Access Control*,

Association for Computing Machinery: Fairfax, Virginia, USA, 1997, pp. 13-19.

[16] W. Wang, "Team-and-role-based organizational context and access control for cooperative hypermedia environments," in *Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia: Returning to our Diverse Roots,* Association for Computing Machinery: Darmstadt, Germany, 1999, pp. 37-46.

[17] M. J. Moyer, and M. Abamad, "Generalized role-based access control," in *Proceedings 21st International Conference on Distributed Computing Systems*, 2001.

[18] E. Bertino, P. A. Bonatti, and E. Ferrari, "TRBAC: A temporal role-based access control model," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 191-233, 2001.

[19] C. K. Georgiadis, et al., "Flexible team-based access control using contexts," in *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies*, Association for Computing Machinery: Chantilly, Virginia, USA, 2001, pp. 21-27.

[20] N. A. T. Tran, and T. K. Dang, "A novel approach to fine-grained content-based access control for video databases," in *18th International Workshop on Database and Expert Systems Applications (DEXA 2007)*, 2007.

[21] A. A. E. Kalam, et al., "Organization based access control," in *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, 2003.

[22] E. Bertino, A. C. Squicciarini, and D. Mevi, "A fine-grained access control model for web services," in *IEEE International Conference on Services Computing, 2004 (SCC'2004). Proceedings. 2004*, 2004.

[23] J. B. D. Joshi, et al., "A generalized temporal role-based access control model," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 1, pp. 4-23, 2005.

[24] N. Yang, H. Barringer, and N. Zhang, "A purpose-based access control model," in *Third International Symposium on Information Assurance and Security*, 2007.

[25] L. Hung, S. Lee, and H. Lee, *Activity-based Access Control Model to Hospital Information*. 2007, pp. 488-496.

[26] D. R. Kuhn, E. J. Coyne, and T. R. Weil, "Adding attributes to role-based access control," *Computer*, vol. 43, no. 6, pp. 79-81, 2010.

[27] F. Kerschbaum, "An access control model for mobile physical objects," in *Proceedings of the 15th ACM Symposium on Access Control Models and Technologies*, Association for Computing Machinery: Pittsburgh, Pennsylvania, USA, 2010, pp. 193-202.

[28] A. Karp, H. Haury, and M. H. Davis, "From ABAC to ZBAC: The evolution of access control models," *ISSA (Information Systems Security Association) Journal*, vol. 8, pp. 22-30, 2010.