A New Modeling Paradigm for Dynamic Authorization in Multi-Domain Systems

Manoj Sastry[1], Ram Krishnan[2], and Ravi Sandhu[3]

[1] Corporate Technology Group, Intel Corporation
manoj.r.sastry@intel.com
[2] Corporate Technology Group, Intel Corporation and George Mason University
rkrishna@gmu.edu
[3] George Mason University and TriCipher Inc.
sandhu@gmu.edu

Abstract. The emergence of powerful, full-featured *and* small form-factor mobile devices enables rich services to be offered to it's users. As the mobile user interacts with multiple administrative domains, he acquires attributes from these interactions. Service providers can tailor services by interpreting user's attributes dynamically at runtime. Such dynamic usage scenarios where attributes from one domain are interpreted and used in another domain motivate the need for dynamic authorization at the time of interaction.
In this paper, we investigate the multi-domain requirements presented by these usage scenarios and explore a new paradigm for modeling these requirements. We examine and extend the UCON model for Usage Control [5] to address the dynamic aspects of multi-domain interactions. The UCON model for usage control is a new foundation of access control which combines traditional authorization with obligations and conditions, mutability of attributes and continuity of decisions. An important observation we make is that attributes, obligations and conditions in UCON are pre-defined. We argue that our multi-domain interaction requirements motivate us to model every UCON component as a dynamic entity. We outline an extended UCON model to accommodate the identified requirements.

1 Introduction

The advent of small form-factor high performance computing devices and high bandwidth ubiquitous networks is enabling users to be connected anytime, anywhere with access to rich, real-time applications. Moreover, as these computing devices become context-aware, they enable applications to dynamically adapt according to the operating environment of the user. As users become increasingly mobile, they transcend multiple security domains.[4] For instance, as Alice moves from a coffee shop domain to a bookstore domain, context acquired in the coffee shop can be interpreted and used at the bookstore for access decisions.

[4] For simplicity, we will often abbreviate 'security domain' simply as 'domain'.

Our objective in this paper is to investigate emerging usage models involving multiple domains, identify their dynamic properties and explore a new paradigm for modeling these properties. Traditional attribute-based access control models have two major limitations: a. In a single domain setting, attributes are typically pre-defined. b. In a multi-domain setting, such models require extensive a-priori agreement of attribute semantics across these systems. We use the term Dynamic Authorization in this paper to collectively refer to the components required for supporting just-in-time authorization. UCON model for Usage Control is an existing robust framework supporting authorization in a single domain system. We propose extensions to the current UCON model to accommodate the requirements of dynamic, multi-domain systems.

The remainder of the paper is organized as follows. Section 2 identifies the properties of a multi-domain system using a coffee shop scenario. Section 3 introduces the new paradigm for modeling Dynamic Authorization. Section 4 gives background of the UCON model. Section 5 describes how the existing UCON model can be extended to accommodate Dynamic Authorization. Section 6 lists related work in this space. Finally, in section 7, we provide a glimpse of some future areas of research and conclude.

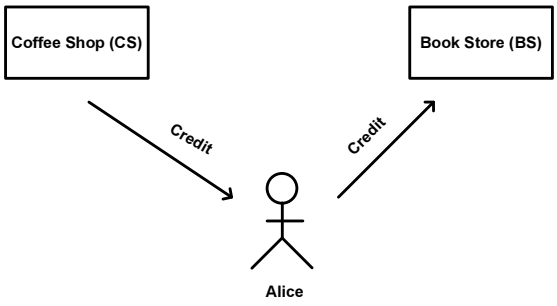

Fig. 1. Coffee Shop Example.

2 Characteristics of Multi-Domain Interactions

In this section, we identify some of the desirable characteristics for user interactions with multi-domain systems. We illustrate these characteristics with a concrete example. Alice walks into a Coffee Shop (CS) and engages in a transaction worth $100. The CS provides a 'credit' worth $10 towards this purchase as an appreciation of this transaction. This 'credit' could be used at various other stores like the Bookstore (BS). Alice later uses this 'credit' towards purchasing a book at the BS. Figure 1 illustrates this scenario. In this example, 'credit' is the context acquired by Alice from the CS and this affects access decision at the BS. We now identify key characteristics of multi-domain interactions using this coffee shop example.

1. Multi-domain interactions: Subjects and Objects interact with multiple systems and this is a key characteristic of mobile systems and applications. In the example, Alice is a subject who interacts with the CS and the BS which are administratively different security domains. Similarly, the objects that Alice carries could be part of these interactions with multiple systems.
2. Information could be dynamic and transcend systems: Due to mobility, information may move from one system to another and could affect access decisions at other systems. In the CS example, Alice obtained a 'credit' from the CS system and used it to purchase a book from the BS system.
3. No prior configuration: In order to interpret information across multiple domains, systems may have to exchange semantics of this information. But in mobile scenarios, information may be dynamically created and hence its semantics cannot be agreed upon a-priori amongst all the systems. It must be interpreted at authorization time. In the CS example, the BS interprets the meaning of 'credit' as a dollar value (by interacting with the CS) just when Alice uses it to buy a book. Prior exchange of 'credit' semantics between the BS and CS will not work because the CS may give a different incentive to customers at different times. For example, the CS may give another customer, say Bob, a 'coupon' whose semantics may not be a dollar value like 'credit'. 'Coupon' could simply be an attestation that Alice made a purchase at the CS.

   In addition to these key characteristics, the following characteristics are desirable in multi-domain scenarios:
4. Support for preserving privacy: Privacy of subject's interactions is an important consideration. In the CS example, Alice may not want the CS to know where she used the 'credit'; she may not want the BS to know where she obtained the 'credit' from, etc. A subject may or may not accept information from a system. Further, a subject may have an option to expose this information or not to other systems. The subject could also completely remove the information that she might have received earlier. [5] From the example, Alice may reject the 'credit' she was given by the CS. On the other hand she may accept the 'credit' but may not expose it to the BS. [6]
5. No prior registration required: In mobile scenarios, subjects may not be pre-registered with a system for interaction. Further, a system need not remember prior interactions with a subject. Thus Alice may not be registered with either the CS or the BS earlier and the CS may or may not remember Alice's interactions the next time she visits the CS.
6. Information could be transferred: Subjects may be allowed to share information she received from a system with another subject (policy permitting). For example, Alice may be allowed to transfer $5 (from her $10 'credit') to a different subject, say Bob. Thus both Alice and Bob would now have a 'credit' worth $5 that could be used at the BS or at other systems.

[5] Note that these requirements could be policy dependent. Also subjects may not be allowed to modify the information she received.

[6] However, if Alice decides to accept the 'credit', she cannot modify the value of this 'credit'. Alice may also decide to remove this 'credit' completely.

This coffee shop scenario will be used as our primary running example throughout this paper. We will also use a few other examples such as airport security systems, health systems, etc., in the later sections for illustrative purposes.

3 New Modeling Paradigm for Dynamic Authorization

In the earlier section, we discussed various characteristics of multi-domain interactions. Our new paradigm is to propose modeling requirements for the three key characteristics:

1. Multi-domain interactions
2. Information could be dynamic and transcend systems
3. No prior configuration

We believe that these three characteristics are missing from current approaches to single-domain and multi-domain authorization systems. Characteristics 1 and 2 require a notion of "Multi-Domain Attributes" which are attributes that needs to be interpreted across multiple domains. However, characteristic 3 requires a notion of "Dynamic Attributes" which are created dynamically and are not pre-defined. In the coffee shop scenario, the 'credit' attribute was dynamically created as an incentive by the coffee shop just for that day when Alice interacted with the system. For this reason, the bookstore cannot write authorization policies to use 'credit' ahead of time. The bookstore needs to dynamically interpret the semantics of 'credit' just when Alice uses it to buy a book. Here 'credit' is also an attribute that can be used at multiple domains (CS and BS). Thus it is a dynamic, multi-domain attribute. Note that Dynamic Attributes are new-born attributes (name-value) as opposed to the notion of *attribute value changing dynamically.*

4 UCON Background

In this section we give a brief overview of the Usage Control (UCON_{ABC}) model [5]. A UCON system consists of six components as shown in Figure 2: subjects and their attributes, objects and their attributes, generic rights, authorizations, obligations, and conditions, where authorizations, obligations and conditions are the components of usage control decisions. An attribute is regarded as a variable with a value assigned to it in each system state. Authorizations are predicates based on subject and/or object attributes, such as role name, security classification or clearance, credit amount, etc. Obligations are actions that are performed by subjects or by the system. For example, playing a licensed music file requires a user to click an advertisement, and downloading a white paper requires a user to fill out a form. Conditions are system and environmental restrictions such as system clock, location, system load, system mode, etc. In UCON, a complete usage process consists of three phases as shown in Figure 3: before-usage, ongoing-usage, and after-usage. The control decision components are checked

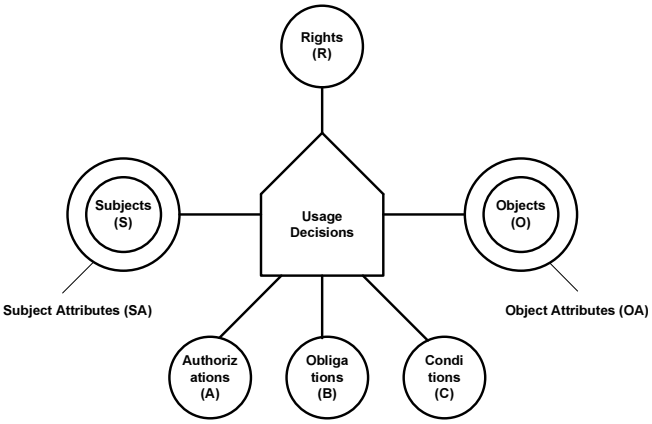

Fig. 2. UCON Components.

and enforced in the first two phases, called pre-decisions and ongoing-decisions respectively, while no decision check is defined in the after-usage phase (since there is no control after a subject finishes an access on an object). The presence of ongoing decisions is called continuity in UCON. Another important property of UCON is attribute mutability. Mutability means that one or more subject or object attribute values can be updated as the results of an access. Along with the three phases, there are three kinds of updates: pre-updates, ongoing updates, and post-updates. All these updates are performed and monitored by the security system. The updating of attributes as side-effect of subject activity is a significant extension of classic access control where the reference monitor mainly enforces existing permissions. Changing subject and object attributes has impact on the future usage of permissions involving this subject or object. This aspect of mutability makes UCON very powerful. For each decision com-

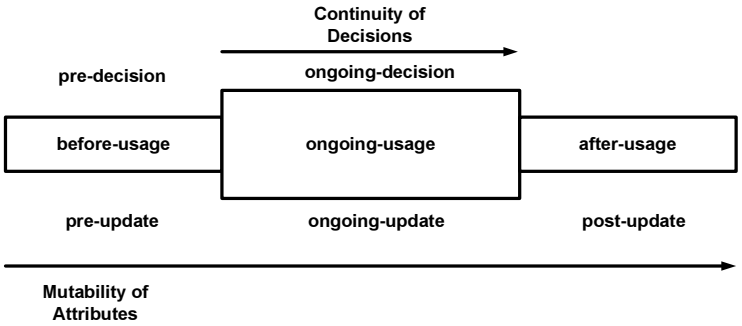

Fig. 3. Continuity & Mutability Properties of UCON.

ponent (authorizations, obligations, and conditions) in UCON, a number of core models are defined based on the phase where usage is checked and updates are performed. Applicable pre and on-going authorization, obligation and condition core models have been defined. The UCON$_{ABC}$ has proven to be robust and highly flexible.

5 The Extended UCON$_{ABC}$ model for Dynamic Authorization

In this section, we examine the major components of the UCON model: attributes, obligations, conditions and authorizations. We enhance the model to accommodate multi-domain interactions. Figure 4 shows a new UCON model with extended components. We call this the extended UCON model abbreviated as EUCON. In the following subsections, we explore each of the EUCON components in detail to support dynamic authorization.

5.1 EUCON Attributes

In UCON, attributes are properties of subjects and objects which are used for usage decisions. In section 3, we introduced the notion of multi-domain and dynamic attributes. In this section, we investigate and classify EUCON attributes.

We can classify attributes based on time at which an attribute is defined:

- *Pre-defined Attributes*: Pre-defined Attributes are similar to the conventional notion of attributes. The semantics of these attributes are defined by the administrator when a system is initially configured.
- *Dynamic Attributes*: Dynamic Attributes are attributes that are defined just-in-time. In the coffee shop scenario, the CS system might define new kinds of incentives like 'credit' at different times on different days dynamically. For example, on a different day the CS could create a 'coupon' attribute which has a different meaning than a dollar value like 'credit'.

We can also classify attributes based on scope as follows.

- *Local Attributes*: Local Attributes are attributes whose semantics can be interpreted only within a single domain and whose values are only visible within that domain. In other words, a Local Attribute has no meaning or visibility anywhere outside the system in which it is defined. In the coffee shop scenario, the the coffee shop may have a Local Attribute called 'id' which may have no meaning outside the CS system.
- *Multi-domain Attributes*: Multi-domain Attributes are attributes whose semantics can be interpreted across multiple domains. In the coffee shop scenario, the book store was able to interpret the semantics of 'credit' that was issued by the coffee shop.

This classification gives us four possible combinations as follows.

– *Pre-defined Local Attributes (PLA)*: PLA's are exactly the same as how current attribute-based models (including UCON) defines attributes. Traditionally, PLA's have served the purpose of access control in a single system (or single domain).
– *Pre-defined Multi-domain Attributes (PMA)*: Current approaches to access control in distributed systems have the notion of PMA's. The approach is to have a clear agreement on the semantics of attributes across all the domains *a-priori*. This is clearly not flexible. It requires extensive configuration beforehand across multiple domains and is not suitable for dynamic scenarios.

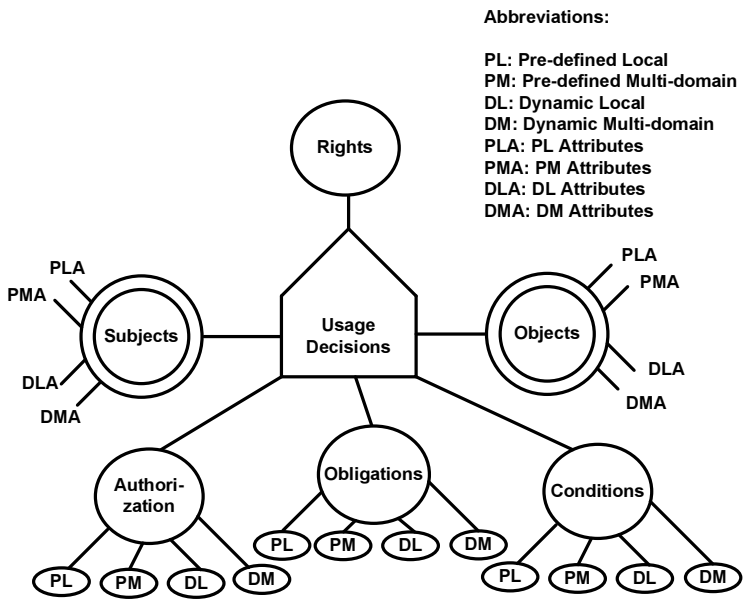

Fig. 4. Extended UCON Components.

– *Dynamic Local Attributes (DLA)*: The notion of DLA's allow systems to dynamically create attributes interpretable within the same system. Typically such an action is deemed as an administrative task. However, we believe emerging next-generation applications (like context-aware applications and such) would demand DLA's. In the coffee shop scenario, on a different day the system may create a new 'discount' attribute that could be used by Alice in the coffee shop itself in the future. This 'discount' may not exist all the time. Here 'discount' is a DLA. Note that DLA's may or may not be persistent.
– *Dynamic Multi-domain Attributes (DMA)*: DMA is fundamentally a new approach to modeling emerging usage scenarios. Here systems may define attributes dynamically that needs to be interpreted at multiple domains.

Prior configuration across multiple domains in such cases does not work because new attributes are dynamically created and other systems may not be able to write policy statements ahead of time. This requires authorization policies to be created dynamically.

The coffee shop scenario gives a clear picture of DMA. On the day Alice interacted with the coffee shop, a new attribute called 'credit' was dynamically created in the system. Since Alice purchsed stuff on that day, she obtained the 'credit' attribute. Further, Alice was able to use this attribute at the bookstore. The bookstore dynamically interpreted the semantics of 'credit' by interacting with the coffee shop and authorized purchasing a book with 'credit'. Thus 'credit' is a DMA.

Dynamic Multi-domain Attributes (DMA) could apply to both subjects and objects. In the coffee shop scenario, it is clear that 'credit' is the DMA of a subject (Alice). Here are few other scenarios to appreciate the generality of DMA's for subjects:

Airport Security: In airport scenario, a passenger interacts with multiple systems and each system may be administratively different. Further each system (security, shops, airlines, etc.) may define their own attributes dynamically. For example, suppose that the security check-in system in an airport and the airline systems are multi-domain systems with no a-priori configuration. When Alice checks-in through the security system, she obtains a DMA called "cleared=true". This DMA could then be used by Alice at the airline's boarding system to board the airplane.

Shopping Mall: In a shopping mall scenario, there are a number of systems (shops, restaurants, parking, etc.). Suppose that Alice dines at a restaurant in the mall. The restaurant offers free parking for their customers. Alice may obtain a DMA "dine@restaurant=true" from the restaurant. She could then use this DMA at a Parking Ticket Validation system for free parking.

Following are few examples of DMA's for objects:

Airport Security: Consider the same airport security example discussed for subjects. When Alice checks in through airport security, all the objects that she carries (e.g. luggage, laptop, etc.) could obtain a DMA "cleared=true". Later on, this DMA could be used by Alice at the airline system in order to carry her objects on the airplane.

Patient's Health Record: Patients (Subject) own their Health Record (Object). As doctors or hospitals use this health record, they may create multiple DMA's on this health record on a daily basis. Examples could be "last viewed", "list of doctor names who viewed the health record", "comments", etc. These DMA's may then need to be interpreted across different hospitals and also at the pharmacy.

Digital Drivers License: The Police systems could define their own DMA on the drivers license object. For example, violations, fines, tickets etc. These DMA's might then be used at other systems to pay the file and also at automobile insurance systems.

5.2 EUCON Authorizations

In UCON, the Authorization component contains rules based on subject and object attributes. We discussed that attributes could be Pre-defined Local Attributes, Pre-defined Multi-domain Attributes, Dynamic Local Attributes and Dynamic Multi-domain Attributes. Because authorization involves constructing rules based on subject and object attributes, we have a similar notion for EUCON authorizations as follows.

- *Pre-defined Local Authorization*: These are rules that are exactly the same as current UCON's definition of Authorization. These rules would serve traditional single system models.
- *Pre-defined Multi-domain Authorization*: This involves constructing rules based on Pre-defined Multi-domain Attributes. Current approaches to authorization in multi-domain systems take this approach. Attributes are predefined and authorization rules are constructed at multiple domains based on these pre-defined attributes.
- *Dynamic Local Authorization*: This involves constructing rules based on Dynamic Local Attributes. In the coffee shop scenario, a dynamic local authorization rule could be constructed (as result of the dynamic local attribute) so that subjects who obtained 'credit' cannot obtain another incentive say 'coupon' at the same time. Alice had a transaction with the coffee shop and thus obtained the 'credit' attribute. The dynamic local authorization rule would prevent Alice from obtaining the 'coupon' at the same time.
- *Dynamic Multi-domain Authorization*: This involves constructing authorization rules dynamically by interpreting the semantics of Dynamic Multi-domain Attributes. In the coffee shop scenario, Alice uses the dynamic multi-domain attribute 'credit' at the bookstore. The bookstore needs to interpret the meaning of 'credit' dynamically and hence construct dynamic multi-domain authorization rules. Exactly how such policies are constructed is an enforcement level issue and restrictions should not be made in the policy model layer.

5.3 EUCON Obligations

In UCON, obligations are actions a subject needs to perform before an access can be granted. For example, a subject may be obligated to 'agree' to a license before access to the target object could be granted. Similar to attributes, we can classify EUCON obligations based on scope and time at which it is defined: local and multi-domain obligations; pre-defined and dynamic obligations.

Local obligations are obligations that can interpreted within a single system (or single domain). Multi-domain obligations have a different notion of obligation from the traditional UCON notion of obligation. A multi-domain obligation is an obligation that a subject needs to perform *in order to use a Multi-domain Attribute*. In the coffee shop scenario, Alice might be obligated to perform some action at the bookstore in order to use the multi-domain attribute 'credit' at the bookstore.

Pre-defined obligations are obligations that are defined when the system was initially configured by an administrator. Dynamic obligations are new obligations that are defined at run-time.

Similar to attributes, we have four combinations: Pre-defined Local Obligations, Pre-defined Multi-domain Obligations, Dynamic Local Obligations and Dynamic Multi-domain obligations. The motivation and necessity of each of these should be clear from previous discussion for attributes. We only give a brief description below.

- *Pre-defined Local Obligations*: These are obligations that are interpretable within a single system and are pre-defined in the system. These are exactly the same as how current UCON model defines Obligations and has proven to useful for traditional single system models.
- *Pre-defined Multi-domain Obligations*: These are pre-defined obligations but are interpretable across multiple systems. Note that these are pre-configured obligations on using Multi-domain Attributes at different systems.
- *Dynamic Local Obligations*: These are obligations that are interpretable only within a single system but are defined dynamically. In the coffee shop scenario, one can imagine new obligations dynamically defined for providing incentives as and when different customers interact with the system. For example, the coffee shop may offer free music to Alice. New obligations could be defined at the time of access. Alice may be obligated to keep an advertisement window open when she accesses music. But the coffee shop may remove this obligation on weekends or define new obligations for using music at different times.
- *Dynamic Multi-domain Obligations(DMO)*: These are obligations defined dynamically and are interpreted at multiple systems at authorization time. Again note that these are obligations on using Multi-domain Attributes at different systems. Consider the coffee shop scenario where Alice uses 'credit' from the Coffee Shop (CS) at Bookstore (BS). Suppose that there are two coffee shops: the coffee shop that issued 'credit' – CS and a coffee shop located within the book store – CS@BS. When Alice uses her 'credit' issued by CS at BS, there could be an obligation that Alice needs to engage in a transaction with the CS@BS before 'credit' could be used at the BS. This is a dynamic multi-domain obligation because the BS discovers the obligation at authorization time and the obligation is on using the multi-domain attribute 'credit'.

5.4 EUCON Conditions

In UCON, conditions are system level factors that need to hold for access to be granted. For example, a server's load should be below a threshold value in order to accept new client connections. Similar to attributes, in EUCON we can classify conditions based on scope and time at which it is defined: local and multi-domain conditions; pre-defined and dynamic conditions.

Local conditions are conditions that can be interpreted within a single system (or single domain). Multi-domain conditions have a different notion of condition

from the traditional UCON notion of condition. A multi-domain condition is a condition that needs to hold *in order to use a Multi-domain Attribute*. In the coffee shop scenario, the bookstore might discover a system level condition that needs to hold when Alice tries to use 'credit'.

Pre-defined conditions are conditions that are defined when the system was initially configured by an administrator. Dynamic conditions are new conditions that are defined at run-time.

Similar to attributes, we have four combinations: Pre-defined Local Conditions, Pre-defined Multi-domain Conditions, Dynamic Local Conditions and Dynamic Multi-domain conditions. Again, the motivation and necessity of each of these should be clear from previous discussion for attributes. We only give a brief description below.

- *Pre-defined Local Conditions*: These are conditions that are interpretable within a single system and are pre-defined in the system. These are exactly the same as how current UCON model defines Conditions and has proven to useful for traditional single system models.
- *Pre-defined Multi-domain Conditions*: These are pre-defined conditions but are interpretable across multiple systems. Note that these are pre-configured conditions on using Multi-domain Attributes at different systems.
- *Dynamic Local Conditions*: These are conditions that are interpretable only within a single system but are defined dynamically. In the coffee shop scenario, Alice might get free access to music within a store. But when she accesses music, a new condition could be defined that says only 10 customers can access music at the same time. On weekends, this condition could be removed and a new condition could be defined the following week.
- *Dynamic Multi-domain Conditions(DMC)*: These are conditions defined dynamically and are interpreted at multiple systems at authorization time. Again note that these are conditions on using Multi-domain Attributes at different systems. Following the coffee shop example for Dynamic Multi-domain Obligations, say that Alice fulfills her obligation. The bookstore could then dynamically discover a condition on using 'credit' that current 'credit' usage on all coffee shop systems has not exceeded $1000 and the 'credit' expires on 01-15-2007. Note that this is a condition on using the dynamic multi-domain attribute 'credit' and the semantics of the condition are interpreted dynamically. Hence this is a Dynamic Multi-domain Condition.

5.5 Discussion

The coffee shop scenario illustrated the need for dynamic multi-domain components in EUCON. New attributes (for subjects and objects), obligations and conditions are to be created at run-time in the system in order to support dynamic and multi-domain components discussed earlier.

Example: In the coffee shop scenario, when Alice interacted with the system, a dynamic multi-domain attribute called 'credit' for Alice was created. Note that Alice never had this attribute prior to this interaction and further this 'credit'

could be used at the bookstore. Dynamic Conditions and Obligations on using this 'credit' could also be defined.

In the standard UCON model, creating new attributes, conditions and obligations is an administrative action and hence should be part of an administrative model. However, from this example, it is clear that for dynamic and multi-domain systems these components needs to be defined at run-time and hence creating such new components should not be part of an administrative model for EUCON. The local components still would remain part of an administrative UCON model but, the dynamic components would be part of the EUCON model itself.

The dynamic components also bring in new EUCON predicates. In figure 4, the subjects and objects have a *set* of attributes. If new attributes are to be created, we need a new predicate to update this set.

For this discussion, we use the examples of dynamic multi-domain attributes, obligations and conditions that were discussed earlier. We provide only an outline for modeling this coffee shop scenario and hence have kept it semi-formal.

We use a few abbreviations for discussion below:
DSA: Dynamic Subject Attributes (local or multi-domain)
DOA: Dynamic Object Attributes (local or multi-domain)
DO: Dynamic Obligations (local or multi-domain)
DC: Dynamic Conditions (local or multi-domain)
CS: Coffee Shop
BS: Book Store

The following are Local Attributes for the Subject (Alice) and the Object (book) within the BS system. *id* represents Alice's identity in the bookstore and *price* is the price of the book that Alice likes to purchase:
$LocalAtt(S) \supseteq \{id\}$
$LocalAtt(O) \supseteq \{price\}$

The CS dynamically defines obligations associated with the dynamic attribute 'credit'. OBS is the obligatory subject in question (in this case S is Alice). OBO is the obligatory object in question (in this case its the coffee shop at bookstore). OB is the obligation itself (in this case it is to $transact$). $getPreOBL$ is the current UCON predicate that obtains the obligation that "Alice (S) is obligated to do a transaction ($transact$) with the coffee shop located in bookstore ($CS@BS$) in order to use her 'credit' to buy a book from the BS ($buywithCredit$)".
$OBS = S$, $OBO = CS@BS$, $OB = transact$
$getPreOBL(S, buywithCredit, O) = (OBS, OBO, OB)$

The CS also dynamically defines conditions associated with 'credit'. $getPreCON$ is the current UCON predicate that obtains the condition that Alice (S) can buy a book (O) using the 'credit' attribute ($buywithCredit$) as long as current credit usage for the coffee shop ($todaysCreditUsageforCS$) is less than \$1000 and the 'credit' has not expired.
$getPreCON(S, buywithCredit, O)) = \{todaysCreditUsageforCS \leq \$1000 \wedge DATE \leq 01-15-2007\}$

When Alice uses this 'credit' attribute to buy a book at the BS, a usage decision is made based on dynamic multi-domain authorization, obligations and conditions. Dynamic multi-domain Authorization predicates are constructed based on the semantics of 'credit'. Dynamic multi-domain obligations and conditions associated with the usage of 'credit' are interpreted by the BS and is used for the usage decision. Again, how such interpretations are made is an enforcement level issue. The *allowed* statement is a usage decision statement that evaluates the authorization, obligations and conditions. Below, we use DA(S) to refer to the set of dynamic multi-domain attributes of subjects:
$allowed(S, buywithCredit, O) \Rightarrow preFulfilled(getPreOBL(S, buywithCredit, O))$
$\wedge\, preConChecked(getPreCON(S, buywithCredit, O)) \wedge$
$DA(S).credit >= price(O) \wedge id(S) =$ "$alice1$"
$preUpdate(DA(S).credit) : credit' = credit - price(O)$

New predicates $AttSetPreUpdate$, $AttValPreUpdate$ and $AttTypePreUpdate$ are used for creating new attributes. When Alice uses the dynamic multi-domain attribute 'credit' from CS at the BS, the BS could create a DSA for Alice called 'coupon' that she may use at other systems:
$AttSetPreUpdate(DA(S)) : DA' = DA \cup \{coupon\}$
$AttValPreUpdate(DA(S).coupon) : coupon' = \$10$
$AttTypePreUpdate(DA(S).type) : type' =$ "$DynamicMultidomain$"

Note that the notion of pre, ongoing and post updates of attributes and pre and ongoing authorizations, obligations and conditions applies here similar to the standard UCON model.

6 Related Work

Many studies have been done in the past on access control in an open and distributed environment. Our approach is a major paradigm shift in modeling information as subject and object attributes that transcends multiple domains and hence needs to be interpreted dynamically. We demonstrated that such an approach facilitates modeling and specifying policies for many current and future usage scenarios in mobile and context-aware applications.

In [2], the authors identify requirements for access control in open environments similar to the ones identified in this paper. They survey extensions that have been proposed in general for models like attribute based and semantics aware access control. However our modeling paradigm of creating and interpreting attributes dynamically across multiple systems is substantially different.

A Contextual Attribute-Based Access Control model (CABAC) is proposed in [1]. The central idea of this paper is that access decisions in contextual application scenarios are made entirely based on attributes. The authors propose to specify authorization policies entirely based on attributes that does not involve either the subjects or objects. They also define Transaction Attributes which are attributes that a subject obtains from a transaction. These Transaction Attributes would fall under our Pre-defined Multi-domain category.

In [4], an authorization framework based on standards like XACML and SAML for distributed systems is proposed. In [7], a UCON authorization framework for collaborative applications like Grids is proposed. In [6], an architecture for secure, independent, interworking services (Oasis) is proposed. Here, clients are authenticated based on roles and access to services are controlled based on proof rules which may refer to multiple services. In [3], an access control mechanism for systems that span multiple administrative domains call dRBAC is proposed. dRBAC is a distributed RBAC that controls activities based on roles and allows delegation of roles across domains.

7 Conclusion and Future Work

In this paper, we explored a new paradigm for modeling dynamic authorizations in multi-domain systems. Current access control models pre-define their components and we demonstrated with compelling usage scenarios that such static definitions would not serve the needs of mobile and dynamic multi-domain interactions. We proposed extensions to the UCON model to express dynamic authorization policies. We identified that attributes, authorizations, obligations and conditions in UCON needs to be dynamic for supporting multi-domain and mobile scenarios. We classified these attributes, authorizations, obligations and conditions based on time of definition as well as their scope. We are in the process of formalizing the notion of dynamic and/or multi-domain components of UCON. A complete and final model would be useful for constructing policies for multi-domain interactions in mobile scenarios.

Many exciting possibilities exist for future work in this area. A formal Extended UCON model for multi-domain interactions needs to be specified. This Extended UCON model should also support many of the characteristics identified in section 2. We hypothesized in section 3, that this extended model would accommodate all the requirements including privacy and this needs to be verified with the formal model. Privacy is an important requirement and in may cases subjects should be able to accept, reject, and delete these dynamic/multi-domain attributes. Subjects should also be able to transfer attributes to another subject. New UCON predicates to support these features have to be studied.

8 Acknowledgments

The authors would like to thank Michael J. Covington for his contribution to early discussions of this paper. This research is partially supported at GMU by grants from NSF and Intel.

References

1. M. Covington and M. Sastry. A contextual attribute-based access control model. *Second International Workshop on Context-Aware Mobile Systems, LNCS 2006*, November 2006.

2. E. Damiani, S. Vimercati, and P. Samarati. New paradigms for access control in open environments. *Proc. of the 5th IEEE International Symposium on Signal Processing and Information*, December 2005.
3. E. Freudenthal, T. Pesin, L. Port, E. Keenan, and V. Karamcheti. drbac: Distributed role-based access control for dynamic coalition environments. *In Proceedings of the Twenty-second IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 411–420, 2002.
4. R. Lepro. Cardea: Dynamic access control in distributed systems. *NASA Advanced Supercomputing (NAS) Division*, 2003.
5. J. Park and R. Sandhu. The ucon abc usage control model. *ACM Transactions on Information and System Security*, 7(1):128–174, Feburary 2004.
6. R.J.Hayton, J.M.Bacon, and K.Moody. Access control in an open distributed environment. *1998 IEEE Symposium on Security and Privacy*, pages 3–14, May 1998.
7. X. Zhang, M. Nakae, M. Covington, and R. Sandhu. A usage-based authorization framework for collaborative computing systems. *Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 180–189, 2006.