Using Spatio-Temporal Role Based Access Control for Physical Access Control Specification: Towards Effective Cyber-Physical Systems

Emsaieb Geepalla, Nassir Abuhamoud

Abstract*— Spatio-Temporal Role-based access control (STRBAC) has been acknowledged as an effective mechanism for specifying access control policies for cyber systems. However, it is not yet clear how a STR- BAC model can be used for specifying access control policies for physical systems. In this paper, we propose a Spatio-Temporal Role Based Access Control (STRBAC) system for modeling the physical access control specification. However, any comprehensive access control model such as STRBAC requires verification mechanisms to ensure the consistency of access control specification. As a result, this paper makes the use of Alloy to perform the analysis of the STRBAC specification. To achieve this, the paper make the use of AC2Alloy to automate the transformation between STRBAC specification and Alloy. With the help of an example, this paper shows how the STRBAC model is transformed into Alloy using our AC2Alloy, and then the produced Alloy model will be analyzed using Alloy Analyzer to detect inconsistencies in the STRBAC specification..

***Index Terms* — spatio-temporal role based access control, alloy, ac2alloy, physical system.**

I. INTRODUCTION

In today's business world, many organizations use information systems to control the access to their information which is available online as we as restrict the access to properties, buildings, zones, rooms, or information resources within the organizations. The need to restrict the access to such a key component of organizations cannot be over emphasized. One of the technologies that organizations have used to achieve this is cyber-physical access control system. However having a correct cyber-physical system is not straightforward. This is because it is very complex system. The complexity comes from the facts that the cyber- physical system should secure the physical world and the cyber world as well as the interaction between both of physical and the cyber processes. As a consequence, many organisation prefer to have two separate systems, one for securing the physical part and the other one for securing the cbyer part.

II. STRBAC FOR PHYSICAL ACCESS CONTROL

It has been found that most of the recent work focuses on modeling and analysis of the cyber access control system [1], [2], [3], [4], [9]. In order to meet the requirements of such systems, several access control models have been proposed such as Role Based Access Control (RBAC) [8] and Spatio-Temporal Role Based Access Control (STRBAC) [9]. Depending on the nature of the organisation a particular access control model is usually more appropriate. In contrast to the most recent work, in this paper we are studying the physical access control system. To do so, we shall make the use of the STRBAC to model the physical access control specification. Spatio-Temporal Role Based Access Control (STRBAC) is one of the access control models that have been found to be very useful in the managing of access rights within organisations, specially when the spatial and temporal information are essential for controlling the access rights. In large organisations such as global enterprises, organisations are usually divided into various regions, departments and zones. Each regions/department/zone has a specific function within the organisation. Staffs of the region/department/zone have responsibilities assigned to them based on the region/department/zone they belong to. Time is also an important factor that should be considered during the access granted process. For example, a user who is a cabling engineer in Essex region can only access to cable chamber in Essex region during his/her normal working hours. With the increasing sizes of organisations and the complexities of job functions within organisations managing this large number of access rights becomes a major problem. In particular, when multiple Access Control policies are com- bined to form new specification, possibly introducing unintended consequences. Such unintended consequences could be introduced due to the existence of in- consistency in the access control specification. The existence of inconsistency in system could pose dangerous security issues that could even cause the downfall of the system. For example, there are many examples where the existence of inconsistency (i.e. wrong user access right) has resulted in huge losses to organisations. It is therefore essential to perform an analysis of STRBAC models to identify inconsistencies in the specification. As a result, lots of work have been presented to identify such inconsistencies using secondary modeling language [3], [17], [19]. Alloy which is a SAT-solver based has been used for analysis of Access control specification. For a small Access Control system the creation of model trans- formation between Access Control model and Alloy could be mange manually, however due to the complexity and size of modern systems an automated trans- formation are required. This is because manually transformation is time con- suming, tedious and error prone. Another issue with manually transformation is the accuracy of the transformation. This means any misinterpretation of the original model will result in incorrect transformation.

Revised Version Manuscript Received on June 10, 2015.

Dr. Emsaieb Geepalla, School of Electronic Engineering, Sebha University, Sebha, Libya.

Dr. Nassir Abuhamoud, School of Electronic Engineering, Sebha University, Sebha, Libya.

In this paper, we propose the use of AC2Alloy [15] which make the use of MDA technique to automate the transformation. This leads to a higher degree of confidence that there is consistency between the two transformations. The security policies of a Physi cal Access Control system will be used throughout the paper to illustrate the transition between STRBAC and Alloy and the analysis that can be applied.

III. PRELIMINARIES

1. SPATIO-TEMPORAL ROLE BASED ACCESS CONTROL (STRBAC).

Several Spatio-Temporal Access Control model have been presented recently to cater for the needs on many mobile application [4], [7], [8], [9]. Our metaphor of Spatial Temporal Access Control is based upon recent work of Inderakshi et al. [10]. There, Access Control is governed by the time and the location conditions, in which the right of assigning a user to a role and permissions owned by that role is banked on spatial and temporal information. In this paper we restrict ourselves to STRBAC without sessions and delegation. If there is no chance of confusion we sometimes use the phrase Access Control instead of STRBAC in the rest of the paper.

1.1. The Basic Concepts of the STRBAC:

The basic concept of the STRBAC model consists of the following five component sets: Users (U), Roles (R), Permissions (P), Times (T) and Locations (L) and the following two relations sets: User Role Assignment (URA) and Permission Role Assignment (PRA).

- U, R, P, T, L are respectively finite sets of users, roles, permissions, times and locations
- User Role Assignment: URA is a relation that associates users with roles based on the time and location, URA $\subseteq$ U $\times$ R $\times$ T $\times$ L. This means users can be assigned to a set of roles at different points of time and location and every role might be assigned to one user or more users at different points of time and location. We write URA(u, r, t, l), meaning that a user u is assigned to a role r at time t and location l.
- Permission Role Assignment: PRA is a relation that associates roles with permissions based on the time and location, PRA $\subseteq$ R $\times$ P $\times$ T $\times$ L. This means roles can be assigned to a set of permissions at different points of time and location and every permission might be assigned to one role or more roles at different points of time and location. We write PRA(r,p,t,l), meaning that a role r is assigned to a permission p at time t and location l.4 STRBAC for Physical Access Control

1.2. Role Hierarchy (RH) in STRBAC:

RH is a partial order on the set ofroles,RH $\subseteq$ R $\times$ R $\times$ T $\times$ L.Wewriter$_i \succcurlyeq$ r$_j$ meaningthattheroler$_i$ isa senior to the role r$_j$ at any time and any location. This means r$_i$ inherits all the permissions of r$_j$, and if there is a user assigned to senior role r$_i$ then he/she could also assign to the junior role r$_j$. RH could be unrestricted, time dependent, location dependent, or time and location dependent and written as $\succcurlyeq$, $\succcurlyeq_t$, $\succcurlyeq_l$ and $\succcurlyeq_{t,l}$ respectively.

1.3. Location Hierarchy (LH) in STRBAC:

LH is a partial order on the set of locations, that specifies which location is outer-location to another location, LH $\subseteq$ L $\times$ L. We write $l_i \succcurlyeq l_j$ meaning that the location l_i is outer-location to the location l_j. This means if there is a user u who has the role r at the outer-location l_i, then this user u should also has the same role r at the inner- location l_j. But we have found that this definition of the Location Hierarchy is not suitable for the physical access control systems as described in section 5.3. Hence, we have modified this definition to meet the requirement of the physical systems as follows.

1.4. Location Hierarchy for Physical System (LHPS):

LHPS is a partial order on the set of locations, that specifies which location is inner-location to another location, LH $\subseteq$ L $\times$ L. We write $l_i \succcurlyeq l_j$ meaning that the location l_j is inner-location to the location l_i. This means if there is a user u who has the role r at the inner-location l_j, then this user u should also has the same role r at the outer-location l_i.

1.5. Separation of Duty between Role (SoDR) in STRBAC:

SoDR is a constraint over roles which specifies the exclusive set of permission, SoDR $\subseteq$ R $\times$ R $\times$ T $\times$ L. We write sodr(r_i, r_j, t, l) meaning that the two exclusive roles r_i and r_j should not be assigned by the same user at time t and location l. SoDR can be unrestricted sodr(r_i, r_j), time dependent sodr(r_i, r_j, t), location dependent sodr(r_i, r_j, l), or time and location dependent sodr(r_i, r_j, t, l).

1.6. Separation of Duty between Permissions (SoDP) in STRBAC:

SoDP it is a constraint over permissions, which specifies the exclusive set of per- mission, SoDP $\subseteq$ P $\times$ P $\times$ T $\times$ L. We write sodp(p_i, p_j, t, l) meaning that the two exclusive permissions p_i and p_j should not be assigned by the same role at time t and location l. SoDR can be unrestricted sodp(p_i , p_j), time dependent sodp(p_i , p_j , t), location dependent sodp(p_i, p_j, l), or time and location dependent sodp(p_i ,p_j, t ,l).

1.7. Cardinality Constraints over Roles (CCR) in STRBAC:

CCR is a constraint over roles, which specifies the restriction on certain roles. We write ccr (r_i, t', l', n), meaning that the role r_i has restriction, so that it should not be assigned by more than n users at time t' and location l'. CCR could be Unrestricted ccr (r_i), Time dependent ccr (r_i,t'), Location dependent ccr (r_i,l'), or Time and Location dependent ccr (r_i, t', l').

IV. ALLOY

Alloy is a language used for modelling and specification of object-oriented sys- tems. It is based on first order logic. Alloy allows analysis of the model via Alloy Analyser, which is SAT-Solver based [5].

An Alloy model consists of a set of modules. Each module consists of one or more paragraphs. The paragraphs of a module can be either signatures, facts, predicates, or check commands. Signatures are used to define new sets of atoms. Indeed a signature is actually more than just a set of atoms of a system, because it could include declaration of relations which depict the relations between such atoms. Those relations will be defined inside the signature body. A signature can also introduce a new set of atoms such as a subset of another set. Constraints such as facts and predicates are used to specify constraints and expressions. A fact is a constraint that always holds and it consists of an optional name, while a predicate is a constraint that can be instantiated in different contexts and has a name. Commands such as check are an instruction to the Alloy Analyser to perform an analysis. A check command helps to search for a counterexample showing that the model is inconsistent. For further details on Alloy and the meaning of elements in Figure 2, we refer the reader to [5].

V. DESCRIPTION OF THE APPROACH

Figure 1 depicts an outline of our approach, which is comprised of two steps. The first step is to use AC2Alloy to convert the Access Control specification to an equivalent specification expressed in the Alloy language. The second step is to use the Alloy Analyzer to analyze the produced Alloy model. These two steps are explained in more detail in the following.

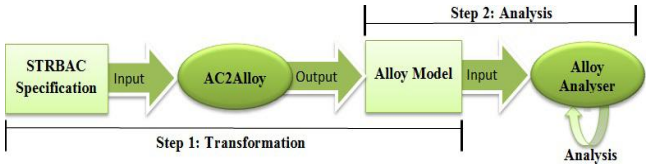

Fig. 1. An outline of our approach

Step 1: Translate the Access Control Specification to Alloy.

To translate the Access Control specification to Alloy we make the use of AC2Alloy. AC2Alloy is an automated tool that make the use of MDA technique (SiTra) to automate the transformation between Access Control and Alloy. When a user provides Access Control specification to the tool AC2Alloy as an input, an XML representation of the Access Control is automatically generated and then the XML representation is automatically transformed into Alloy.
6 STRBAC for Physical Access Control

Step 2: Analysis using the Alloy Analyser.

The procedure defined in the previous step results in the production of an Alloy model of the model transformation. The Alloy Analyser can then be used to analyse the Alloy model. The Alloy Analyser can be used to check whether Alloy checks (certain statements that should hold according to the specification) are satisfied or not. If Alloy Analyser found that a check is not satisfied, then the analyser presents a counterexample, which is an instance of the Alloy model that violates the specification.

VI. EXAMPLE OF SPATIO-TEMPORAL ROLE BASED ACCESS CONTROL

In this section, we describe an example of Spatio-Temporal Role Based Access Control to illustrate the method proposed in this paper, but before that we shall provide a brief description of Physical Access Control.

4.1. Physical Access Control System

Physical Access Control is a system that helps organisations in restricting en- trance to a property, a building, a zone, or a room to authorised users using technologies such as card reader. For example, an Essex cabling engineer try to get access to an cable chamber in Edinburgh, the access may be denied as it is out of his/her working region. However, if there are pre-registered approvals or special events, such as a meeting, emergency repair, or disaster recovery, then the access can still be granted. In a large organization such as a global enterprise, managing such physical access control policies is a complex task. The complexity is mainly from four aspects:

1. A large number of building/zones with distributed geo-locations
2. Buildings with different risk levels, which required different levels of access control.
3. A mixture of users who have access to buildings based on a mixture of roles such as permanent employees, contractors, third parties and outsourcing workforce.
4. Time constraints for accessing the building/zones.

Considering the above factors, an ideal access control system for a large organisation should be able to assess a combination of risk levels of building/zones, personal role, time and location. A typical physical access control scenario is illustrated in Figure 2, and described as following:

– A user swipes ID card to submit his/her user profile for the system assess. The user profile contains the user's role information for access control such as the roles that he/she could assign to and the time and the location information for the user to role assignment.

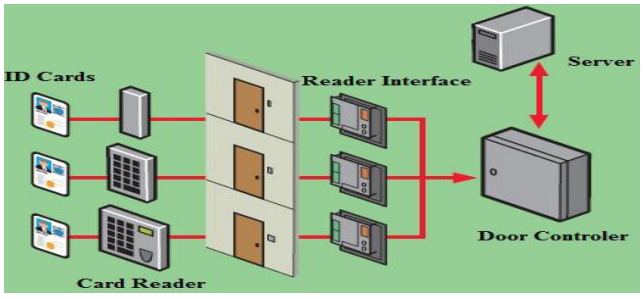

Fig. 2. Physical Access Control System

– A user swipes ID card to submit his/her user profile for the system assess. The user profile contains the user's role information for access control.
– The card reader reveals which building/zone the user is accessing and there- fore, submits the user profile and building profile to the system. The build- ing/zone profile contains the building/zone's risk level information for access control.

– Both profiles are processed by the physical access control system. The process involves a rule engine who has the knowledge of the physical access policies. It compares the two profiles and makes an access decision.
– Three possible results (access, cannot access, or access without approval) can be returned by the access control process. If it returns ”access”, it could be because of one of the following scenarios: direct assignment due to User Role Assignment and Permission Role Acquire, indirect assignment due to Role Hierarchy, or indirect assignment because the user profile has pre-registered approval. If it returns ”access without approval”, then the access control system will look for registered approvals in the user profile for the required zone to decide whether the access is granted or denied. If the rule engine returns ”cannot access”, then the access control is denied.

4.2. Physical System Security Policies

An example of the physical access control policy is as following:

1. User roles are mainly categorised on user’s job type, for example: company employees, technical employees, clerical employees, and cabling engineer.
2. The organisation is based on several regions for example: Birmingham and Manchester and every region consists of several buildings which contain sev-8 STRBAC for Physical Access Control eral zones that are classified based on the risk level for example: medium risk zone is inner-zone t the low risk zone.
3. Technical employees rights and clerical employees rights consist of all rights from company employees within their working region during the DayTime.
4. Cabling engineers rights consist of all rights from technical employees within their working region during the DayTime.
5. Cabling engineers can access street cabinets within their working region during the working hours that is DayTime.
6. Company employees can access low level zone (common rooms and coffee rooms) only at their working region at the working hours that is DayTime.
7. Clerical employee can access medium level zone (data center) only at their working region at the working hours that is DayTime.
8. Same user should not be a clerical employee and technical engineer at the same time and the same Location.
9. The high level risk zone is inner-zone to the medium level risk zone.
10. The medium level risk zone is inner-zone to the low level risk zone.
11. The cabling engineer role should not be assigned by more that one user at the same time and the same location.
12. The organisation consists the thousands of users. Dave, Mark, Sarah, Jenny, James, Hanna, Kate and Amy are a small list of the users within the organisation that has been chosen to illustrate our approach. This list of users could be assigned to the roles in the organisation as illustrated in Table 1.

Table 1. User Role Assignment Constraints

Users	Roles	Times	Locations
Dave	cabling engineer	DayTime	Birmingham
Sarah	cabling engineer	DayTime	Birmingham
Amy	technical engineer	DayTime	Birmingham
Mark	clerical employee	DayTime	Birmingham
James	cabling engineer	DayTime	Manchester
Jenny	technical engineer	DayTime	Manchester
Kate	clerical employee	DayTime	Manchester
-	-	-	-

The above policies could be represented using STRBAC as illustrated in Table 2.

Table 2. STRBAC Policy for Physical Access Control Policy

{Users}={Dave, Mark, Sarah, Jenny, James, Hanna}
Permissions={Access Low Risk Zone Birmingham (ALRZB) , Access Medium Risk Zone Birmingham (AMRZB), Access Street Cabinets Birmingham (ASCB), Access Low Risk Zone Manchester (ALRZM), Access Medium Risk Zone Manchester (AMRZM), Access Street Cabinets Manchester (ASCM)}
Roles={company employees, technical employees, clerical employees, cabling engineer}
Times={DayTime, NightTime}
Locations={Medium risk zone Birmingham, Medium risk zone Manchester, low risk zone Birmingham, low risk zone Manchester, Street Cabinets in Manchester, Street Cabinets in Birmingham, Birmingham, Manchester}
URA={ura(Dave, cabling engineer, DayTime, Birmingham), ura(Sarah, cabling engineer, DayTime, Birmingham),), ura(Amy, technical engineer, DayTime, Birmingham), ura(Mark, clerical employee, DayTime, Birmingham), ura(James, cabling engineer, Day- Time, Manchester), ura(Jenny, technical engineer, DayTime, Manchester), ura(Kate, clerical employee, DayTime, Birmingham)}
PRA={pra(company employees, ALRZB, DayTime, Birmingham), pra(company em- ployees, ALRZB, DayTime, Manchester), pra(cabling engineer, ASCB, DayTime, Birmingham), pra(clerical employee, AMRZB, DayTime, Birmingham), pra(cabling engineer, ASCM, DayTime, Manchester), pra(clerical employee, AMRZM, DayTime, Manchester)}
RH={rh(technical employees ≥ company employees), rh(clerical employees ≥ company employees), rh(cabling engineer ≥ company employees)}
LHPS={lhps(Birmingham ≥ Medium risk zone Birmingham) , lhps(Birmingham ≥ Low risk zone Birmingham), lhps(Manchester ≥ Medium risk zone Manchester), lhps(Manchester ≥ Low risk zone Manchester), lhps(Birmingham ≥ Street Cabinets in Birmingham), lhps(Manchester ≥ Street Cabinets in Manchester)}
SoDR={sodr(technical employees, clerical employees)}
SoDP={sodp(ALRZB, ASCB), sodp(AMRZB, ASCB), sodp(ALRZM, ASCM), sodp(AMRZM, ASCM)}
CC={cc(cabling engineer, 1)}

4.3. Applying our method to the Case Study

AC2Alloy is used to generate an Alloy model from the STRBAC specification in the context of the Physical Access Control system.

As described in section 3, when we enter the STRBAC specification to the tool AC2Alloy, an XML representation of the Access Control is automatically generated and then the XML representation is automatically transformed into Alloy. The basic components of STRBAC model such as sets of Users, Permissions, Times and Locations will be transformed to the following signatures:

```
abstract sig User{}
one sig Amy, Dave, Kate, Mark, Jenny, Sarah, James extends
User{}
abstract sig Location{}
one sig Birmingham, LowRiskZoneBirmingham,
Manchester,10 STRBAC for Physical Access Control
LowRiskZoneManchester,MediumRiskZoneBirminghm,Me
diumRiskZoneManchester,StreetCabinetsInBirmingham,Str
eetCabinetsInManchester extends Location{}
abstract sig Time{}
one sig DayTime, NightTime extends Time{}
abstract sig Permission{}
one sig ALRZB, ALRZM, AMRZB, AMRZM, ASCB,
ASCM extends Permission{}
```

The set of Roles will be transformed into signatures, facts and predicates. The facts and predicates are used to represent the relationships between roles, users, permissions, times and locations. Such relationships are expressed by User Role Assignment (URA) and Permission Role Acquire (PRA). For example AC2Alloy will transform the role Clerical Employees and the User assignment of the role Accounting to the following Alloy code:

```
abstract sig Role {time: lone Time,location: lone
Location,user: set User,permission: set Permission}
one sig ClericalEmployees extends Role{}
fact ClericalEmployees_fact{all self: ClericalEmployees |
ClericalEmployeesCondition[self]}
pred ClericalEmployeesCondition[self:
ClericalEmployees]{((self.permissions=
none)&&(self.location= Birmingham)&&(self.time =
DayTime)&&(self.users = Mark ))||
  ((self.permissions= none)&&(self.location= Manchester)
&&
  (self.time= DayTime)&&(self.users= Kate ))}
```

The Permission Role Acquire injects the predicate within Alloy code for the Roles with new assignment information. For example the transformation of the Per- mission Role Assignment of the role Clerical Employees will inject the predicate ClericalEmployeesCondition within the role Clerical Employees with new assign- ment information as shown in the following Alloy code:

```
pred ClericalEmployeesCondition[self: ClericalEmployees]{
  ((self.permissions= AMRZB)&&(self.location=
Birmingham) &&
  (self.time = DayTime)&&(self.users = Mark ))||
  ((self.permissions= AMRZM)&&(self.location=
Manchester) &&
  (self.time= DayTime)&&(self.users= Kate ))}
```

The Role Hierarchy also injects the predicates within the Alloy code for the Roles with new assignment information. For example the effect of the trans- formation of the hierarchy between the roles Clerical Employees and CompanyTitle Suppressed Due to Excessive Length 11 Employees will be transformed into new assignment information which will be injected to the predicates within the Alloy code for the roles Clerical Employ- ees and Company Employees. For example the senior role Clerical Employees can inherit all the permissions assigned to the junior role Company Employees,

then the predicate ClericalEmployeesCondition within the role Clerical Employ- ees will be injected with new assignment information as shown in the following Alloy code:

```
pred ClericalEmployeesCondition[self: ClericalEmployees]{
  ((self.permissions= AMRZB+ALRZB)&&(self.location=
Birmingham) &&
  (self.time = DayTime)&&(self.users = Mark ))||
  ((self.permissions= AMRZM+ALRZM)&&(self.location=
Manchester) &&
  (self.time= DayTime)&&(self.users= Kate ))}
```

4.4. Model Analysis

In the previous section, we have shown the translation between the Access Con- trol model and the corresponding Alloy. However, this work would be incomplete without demonstrating how we can verify the produced Alloy model to detect inconsistency. In This section, we provide a brief description of the automatic analysis task that carried out via Alloy analyser, but before that we shall intro- duce several common inconsistencies that occur in the Physical Access Control policy.

1. A user has access to an inner-zone, but does not have access to outer-zone.
2. A user has two conflicted roles that are constrained by Separation of Duty between Roles at the same time and the same location.
3. A user has given default access to zones that are outside of his/her work region.
4. A user has given default access to zones that are outside of his/her work region.
5. A user has the right to access two conflicted permissions that are constrained by Separation of Duty between Permissions at the same time and the same location.
6. A role is assigned by a number of users that exceed the maximum number of user that should be assigned to that role at the same time and the same location.

To ensure that the Physical Access Control system is consistent, several Alloy check has been produced via AC2Alloy. For example, An Alloy check will be generated with every Separation of Duty constraint, so that the check could be used to verify whether the constraint is hold or not. An instance of this is that the Separation of Duty between the roles technical employees and clerical employees will be transformed into a predicate and checks as follows:

```
pred SODR[r1, r2:Role, l:Location, t:Time]{all u:User |((u in
r1.users)&&(t in r1.time)&&(l in r1.location)=>12 STRBAC
for Physical Access Control((u not in r2.users)&&(t in
r2.time)&&(l in r2.location))}
sodr1 :check
{SODR[TechnicalEmployees,
```

ClericalEmployees, DayTime, Birmingham]}
sodr1 :check { SODR[TechnicalEmployees, ClericalEmployees, DayTime, Manchester]}
sodr1 :check {SODR[TechnicalEmployees, ClericalEmployees, NightTime, Birmingham]}
sodr1 :check {SODR[TechnicalEmployees, ClericalEmployees, NightTime, Manchester]}

The execution of the all the above Alloy checks shows that Alloy Analyser did not find any counterexample. This means the Separation of Duty statement is hold, so that there is no inconsistency in the Access Control policy cause by this constraint.

Another example of Alloy checks which will be generated via AC2Alloy is the checks which will be generated for the Cardinality Constraint. For example the Cardinality constraint over the role cabling engineer will be transformed into Alloy checks as follows:

cc1 :check {((Birmingham in Cabling Engineer . location) & &(DayTime in Cabling Engineer.time) => (#CablingEngineer.users < 1))}
cc2 :check {((Manchester in Cabling Enginee .location) &&(DayTime in Cabling Engineer.time) => (#Cabling Engineer.users < 1))}

The execution of the first Alloy check shows that Alloy Analyser picked up a counterexample as depicted in Fig 2. This means the policy is inconsis- tent because there is more than one user (Dave and Sarah) are assigned to the role cabling engineer at Birmingham region during the DayTime, which is not permissible according to the Cardinality constraint. The The execution of the second Alloy check shows that Alloy Analyser did not find any counterexample. This means the second cardinality statement is hold and there is no more than one user that has the right to have the role cabling engineer att the location Manchester and during the DayTime.

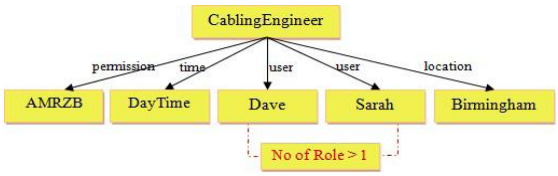

Fig. 3. Counterexample for the Cardinality check cc1

VII. DISCUSSION

In discussing the pros and cons of using STRBAC to formalize Physical Access control specification and using Alloy to analyse the Physical Access Control specification, there are several issues to consider:

- Choice of STRBAC to formalize the physical access control.
- Choice of Alloy for analysis of Physical access control specification.
- Challenges of transformation between STRBAC and Alloy.
- Differences between physical systems and cyber systems.

5.1. Choice of STRBAC to formalise the Physical access control

It has been found that there is shortage of works on formalisation of Physi- cal access control. In [2], Sampemane et al has used Role Based Access control for physical spaces that allow novel uses of physical spaces, while ensuring that resources in these spaces are not misused.

The previously cited work present formalizations that does not support the spatial and temporal constraints. We have given an example above (Section 4) in which the spatial and temporal information are important factors that should be considered before accept or reject the access request. As a result, we have made the decision to make the use of STRBAC which support both spatial and temporal information to model the physical access control specification. The delegation also should be considered before granting or dining the access in physical access control. For example, Birmingham cabling engineer might delegates his permissions to Essex cabling engineer during special event such as emergency repair or disaster recovery. Al- though delegation is not considered in this paper, the STRBAC model [4] sup- ports delegation. Extending our approach to consider delegation remains for future research. Despite the fact that the STRBAC has fulfilled most of the physical system policies, the STRBAC model [4] still has some limitations to meet all the requirements of physical system. An example of the physical access control policies which can not be formalized using the current STRBAC model is pre-requested permission. Pre-requested permission means in the organisations a user who has a specific permission (i.e. access to low risk zone) based on the role assigned to him/her might request to have another permission (i.e. access to medium risk zone) which is not assigned to his role. As a result, we are planning the extend the current STRBAC model to consider such elements.

5.2. Choice of Alloy for analysis of Physical access control specification.

Our work is motivated by the following ideas: Firstly, the Physical Access Control system designer should be able to automatically analyse the system prior to its deployment. Secondly, the increase in the size of organisation has increased the complexity and the size of access control systems, as a result, this has the demand of finding an automated tool for the analysis of access control specification. Alloy [5] which is a SAT-solver based has been used for analysis of the specification of cyber access control system [3], [4], [8] [16]. In contrast with the most recent works, our approach make the use of Alloy for analysis of the specification of physical access control. The choice of Alloy as the formal language for modelling and analysing of Access Control is due to the following reasons:

1. Alloy is supported by a tool called The Alloy Analyser, which provides sup- port for fully automated analysis of Alloy models with the help of SAT solvers.

2. The tool provides the capability to search for counterexamples in the model using commands such as check. A counterexample is an instance of the model that violate the specification of the system [11].

Similar to the most automated tools that have been used for analysis of access control specification, Alloy also has some limitations. One of the major challenges of Alloy is scalability. To deal with such issue, we have created Alloy model for STRBAC with a minimum number of signatures, so that the we could reduce the number of atoms which will be generated by Alloy analyser during the analysis. To achieve this, we have transformed the User Role Assignment, Permission Role Acquire and Role Hierarchy to predicates instead of signatures. As a result the Alloy models produced by the method presented in this paper could scale well.

5.3. Challenges of the transformation between STRBAC and Alloy

One of the main challenges of using second model languages for analysis of access control is the transformation between the source model and the target model. Due to the complexity and size of modern Access Control systems the trans- formation between the source model, STRBAC model and the target model, Alloy model has become challenging and cannot be carried out manually. This is because manually transformation is time consuming, tedious and error prone. Another issue with manually transformation is the accuracy of the transformation. To deal with such issue, our approach makes the use of MDA techniques to automate the transformation between STRBAC model and Alloy model. Our approach is generic and valid for any STRBAC model and not just for the case study presented in this paper. For instance, AC2Alloy transform any STRBAC model to Alloy model to make it amenable to formal analysis using Alloy Anal- yser. The use of model transformation in supporting interoperability between de- sign and analysis models in software engineering is increasingly gaining importance in the software development community. Anastasakis et al [32] describe the challenge of model transformation from UML to Alloy [33]. They propose UML2Alloy [34] as a tool for the analysis of UML models via the Alloy frame- work. UML2Alloy allows the analysis of static models which are qualified with OCL constraints [35]. To automate the transformation between two models, there are several chal- lenges that we have faced. The main challenge was finding efficient ways of trans- lating Access Control policies into compact, manageable models. This is done through a number of transformations described with the help of the example pre- sented in section 4.3. However, our transformation still have some limitations. For example, some elements of the STRBAC model [4] such as delegation is not considered. Delegation is very important elements in many real world systems. As a result, extending the tool AC2Alloy to support such elements will be a topic for future research.

5.4. Differences between the Physical Access Control Systems and the Cyber Access Control Systems

Using the STRBAC for modelling the physical access control specification and the cyber access control specification has shown us that there are some major differences between the cyber system and the physical systems. One of these differences is formalising of Locations Hierarchy. In general Location Hierarchy is a partial order on the set of Locations, that specify which location is outer- location to another location. Next we shall show the differences between the Location Hierarchy in the cyber systems and the physical systems with the help of example. Assume that an organisation is divided into two locations l_i and l_j and the location l_i is outer-location to the location l_j as illustrated in Figure 4.

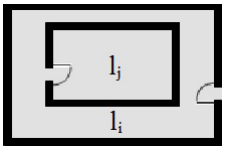

Fig. 4. Location Hierarchy

In the cyber systems, the hierarchy between the two locations l_i and l_j means that if there is a user u who can access to the role r at the outer-location l_i, then this user u should has the same role r at the inner-location l_j. While in the physical systems, the hierarchy between the two locations l_i and l_j means that if there is a user who can access to the role r at the inner-location l_j , then this user u should also access to the same role r at the outer-location l_i. To illustrate the differences between the effects of the Location Hierarchy on the physical systems and the cyber systems we have created the following two tables. Table 3 shows all the possible scenarios that may occur when a user u is trying to access the role r in the physical system. For example, the first row in Table 3 shows that if the user u can access to the role r at the outer-location l_i and the same user u can access to the same role r at the inner-location l_j, then this user u should access to the role r either at the oouter-location l_i or the inner-location l_j. Another example is that, the third row in Table 3 shows that if the user u can not access to the role r at the outer-location l_i and the same user u can access to the same role r at the inner-location l_j, then this user u should access to the role r either at the outer-location l_i or the inner-location l_j based on the Location Hierarchy in the physical systems. Whereas, Table 4 shows all the possible scenarios that may occur when a user u is trying to access the role r in the cyber system. For example, the first row in Table 4 shows that if the user u can access to the role r at the outer-location l_i and the same user u can access to the same role r at the inner-location l_j, then this user u should access the role r either at the oouter-location l_i or the inner-location l_j. Another example is that, the second row in Table 4 shows that if the user u can access to the role r at the outer-location l_i and the same user u can not access to the same role r at the inner-location l_j, then this user u should access the role r either at the outer-location l_i or the inner-location l_j based on the Location Hierarchy in the cyber systems. We believe that identifying such differences between the cyber systems and the Physical systems is very essential task in order to design and implement an effective Cyber-Physical systems. This is because Cyber-Physical systems involve some interaction between the cyber and the physical systems.

Table 3. Physical System

i	lj	l_i and l_j
True	True	True at l_i or l_j
True	False	True only at l_i
False	True	True at l_i or l_j

False	False	False

Table 4. Cyber System

li	lj	li and lj
True	True	True at li or lj
True	False	True at li or lj
False	True	True only at lj
False	False	False

VIII. PERFORMANCE

To evaluate the performance of AC2Alloy tool, a set of case studies including the case study discussed in section 4 were used. The results of these case studies have shown that AC2Alloy could successfully convert these case studies to Alloy in order to be analysed using Alloy Analyser. To evaluate the scalability of AC2Alloy we have started testing the tool using a medium scale access control system. This has shown us that the tool could transform a medium scale access control very quick. Then we have increased the size of the system several time and it has been found that the speed of the transformation start decrease slightly as the size of the system increase. In all cases however, the correct transformation was produced. These experiments were performed on a laptop with a AMD Athlon(tm)x2 dual-core ql-64 2.10 GHZ and 3GB RAM under Windows 7. The unit for time spent is ms (millisecond). Table 3 presents the results for the several testing cases.

Scenario	User	Role	Permission	Time	Location	URA	PRA	RH	LH	SoD	CC	No of all Elements	Speed of Transformation
Scenario 1	50	10	10	5	5	50	25	5	2	10	5	177	150ms
Scenario 2	100	20	20	10	10	100	50	10	4	20	10	354	220ms
Scenario 3	150	30	30	15	15	150	75	15	6	30	15	531	500ms
Scenario 4	200	40	40	20	20	200	100	20	8	40	20	708	2800ms
Scenario 5	250	50	50	25	25	250	125	25	10	50	25	885	1300ms
Scenario 6	300	60	60	30	30	300	200	30	12	60	30	1062	2000ms

Table 5. AC2Alloy Evaluation Result

We have also tested the Alloy code produced by AC2Alloy. To do so, we have created 6 scenarios and computed the time spent on analysis of every scenario as illustrated in table 4. In every scenario in table 4 we have increased the number of one element of the STRBAC model. For example, in the scenario 2 we have increased the number of Users and in the scenario 3 we have increased the number of Roles. Table 4 shows that in all scenarios a counterexample has been found. This means the Alloy analyser has found at lest one instance which is conflicted with the Access Control specification in the all scenarios. Another observation from the table 4 is that the time spent on analysis of Alloy code increases slightly as the number of all elements of STRBAC increase. This is because Alloy Analyser is a SAT-solver based and SAT-solving time may vary enormously depending on factors such as clause ordering, number of variables and average length of clauses [14]. It can also be noted that some elements of the STRBAC model such as Roles effect the speed of analysis more than the other elements. This is because in our model transformation every role is transformed into a Signature, Fact and Predicate. This means the number of variables and clauses which will be created by the SAT-solver for every role is larger than the number of variables and clauses which will be created by the SAT-solver for any other elements such as user or permission. As consequence, when the number of roles increased by N number, the time spent on analysis will be larger than the time spent when any other elements of the STRBAC increased by the same number N.

Scenario	User	Role	Permission	Time	Location	URA	PRA	RH	LH	SoD	No of all Elements	Counterexample	Time Spent on Analysis
Scenario 1	30	10	10	2	5	30	10	2	2	1	102	Found	12.6ms
Scenario 2	300	10	10	2	5	300	10	2	2	1	642	Found	31.4ms
Scenario 3	30	100	10	2	5	300	10	2	2	1	462	Found	39.2ms
Scenario 4	30	10	100	2	5	30	100	2	2	1	282	Found	19.7ms
Scenario 5	30	10	10	2	5	30	10	25	2	1	125	Found	13.1ms
Scenario 6	30	10	10	2	5	30	10	2	25	1	125	Found	13.2ms

Table 6. Alloy code Test Result

IX. CONCLUSION

In this paper, we propose a Spatio-Temporal Role Based Access Control (STR- BAC) model for the specifying access control policies in the physical systems. The paper also proposes to use formal methods for analysis of STRBAC specification. To achieve this, the paper describes a method AC2Alloy that makes the use of Model Transformation techniques to auto-generate of Alloy from STR- BAC specification, thus allowing for powerful analysis to take place using Alloy analyser utilizing SAT-Solvers. The suggested approach has been evaluated with the help of a real-world example.

REFERENCES

1. Gerd Behrmann, Alexandre David, Kim Guldstrand Larsen, John Hakansson, Paul Petterson, Wang Yi, and Martijn Hendriks. UPPAAL 4.0. In Proceedings of the 3rd international conference on the Quantitative Evaluation of Systems, pages 125126, Riverside, California, USA, September 2006.
2. Indrakshi Ray and Manachai Toahchoodee. A Spatio-Temporal Access Control Model Supporting Delegation for Pervasive Computing Applications. In Proceed- ings of the 5th International Conference on Trust, Privacy and Security in Digital Business, pages 48.58, Turin, Italy, September 2008.
3. Arjmand Samuel, Arif Ghafoor, and Elisa Bertino. A Framework for Specification and Verification of Generalized Spatio-Temporal Role Based Access Control Model. Technical report, Purdue University, February 2007. CERIAS TR 2007-08.
4. Manachai Toahchoodee and Indrakshi Ray. On the Formal Analysis of a Spatio- Temporal Role-Based Access Control Model. In Proceedings of the 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security, pages 17.32, London, U.K., July 2008.
5. Jackson.Daniel (2006), Software Abstractions Logic, Language, and Analysis, Cam- bridge: The Mit Press.
6. Gerd Behrmann, Alexandre David, and Kim Guldstrand Larsen. A Tutorial on Uppaal. In 4th International School on FormalMethods for the Design of Computer, Communication and Software Systems, pages 200236, Bertinoro, Italy, September 2004.
7. Liang Chen and Jason Crampton. On Spatio-Temporal Constraints and Inheritance in Role-Based Access Control. In Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, pages 205.216, Tokyo, Japan, March 2008.
8. Hsing-Chung Chen, Shiuh-Jeng Wang, Jyh-Horng Wen, Yung-Fa Huang, Chung- Wei Chen: A Generalized Temporal and Spatial Role-Based Access Control Model. JNW 5(8): 912-920 (2010)
9. Indrakshi Ray and Manachai Toahchoodee. A Spatio-temporal Role-Based Access Control Model. In Proceedings of the 21st Annual IFIPWG11.3Working Conference on Data and Applications Security, pages 211.226, Redondo Beach, CA, July 2007.
10. Basit Shafiq, James B. D. Joshi, and Arif Ghafoor. Petri-net model for verification of RBAC Policies. Technical report, Purdue University, 2002.
11. SamratMondal,ShamikSural,andVijayalakshmiAtluri.TowardsFormal Security Analysis of GTRBAC using Timed Automata. In Proceedings of the 14th ACM Symposium on Access control Models and Technologies, pages 3342, Stresa, Italy, June 2009.