ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (203-217), Month: July - September 2016, Available at: www.researchpublish.com

HOr-BAC: An Access Control Based On Hierarchical Organizational

Benoît Martin AZANGUEZET QUIMATIO[1], Laure Pauline FOTSO[2]

[1]Faculty of Sciences, Department of Mathematics and Computer Science, University of Dschang, Dschang, Cameroon
[2]Faculty of Sciences, Department of Computer Science, University of Yaounde I, Yaounde, Cameroon

Abstract: **The access control models like DAC, MAC, RBAC, TBAC, TMAC or OR-BAC does not permit to define security policies that will enable to control the activities of a super-user or Database Administrator (DBA). Moreover, the super-user has more rights and powers over the information system resources than its hierarchical superiors. This paradox exposes the organisation to attacks targeting the information system. We propose a concept of *electronics signature book* based on an extension of the Or-BAC model, that we developed and called HOr-BAC to specify security policies capable of solving these problems control of super-user activities. We implemented this concept into a Postgres SQL Database System Manager, for controlling the super-user activities in bank information system with success.**

Keywords: **component; access control model, on hierarchical Organization, operational unit, administrative unit, computer as information system, request, validation, treatment mode, electronic signature-book.**

I. INTRODUCTION

The development of information and communication technology has made information process faster and more accurate through faster processors, storage supports, sharing techniques, and improved data transmission [28]. This has encouraged the emergence of computerized information systems (CIS). These CIS however face security access issues to data and treatments. In fact, actual models of access control proposed DAC [17], MAC [11, 3], RBAC [24, 12, 11], TBAC [6], TMAC [7] or OrBAC[22, 14] have focused on traceability and auditability properties of information systems. But these properties do not prove the absolute confidentiality and integrity requirements, requested by the organizations. They have rather concentrated all powers and authorisations at this super-user or DataBase Administrator (DBA) and do not provide any mechanism to implement security policies capable of controlling the super-users activities in the CIS. This leads to a serious violation of the organizational hierarchy because even the general manager who is the superior to the super-user must request to the super-user to access certain data and treatment.

In this paper we propose the concept of *electronic signature-book* which is an automatic control process of emission, processing and request execution in a CIS. The *electronic signature-book* considers any activity (such as *consultation*, *adding* and *modification*) in the system as a request that must be processed by the superior of the emitter before running in the CIS. It uses an extension of the model ORBAC called HOr-BAC that we have developed. Our model HOr-BAC is based on three additional concepts: - *organizational structure* which laminates the organization in *operational units* and in *administrative units*; - *work employees* which are allocated to operational units; - *request* of changing of state of system resources emitted by the work employees; - *treatment mode* of requests defined by the security policies of the organization.

Like the other models, our approach can trace and audited the activities of all users on the system. In addition, it can control the super-user activities. An implementation of our electronic signature book was made with satisfaction, in Postpres SQL DBMS to secure the CIS of a bank agency with 10 employees, connected to the database from remote positions.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (203-217), Month: July - September 2016, Available at: www.researchpublish.com

In the following paper, the sections 2 presents the works related to access control models. Sections 3 deals with the concepts of the organization roles based access control models (Or-BAC). Section 4 describes our model HOr-BAC. Section5 proposes an example of implementation of our electronic signature-book, using the organizational structure of an academic department. Section 6 concludes the paper.

II. RELATED WORK

A. Discretionary access controls models (DAC)

This model is defined by Harrison, Ruzzo and Ullman(HRU) [13]. The model HRU is based on three concepts: *subjects*, *objects* and *actions*. The subject is an active entity including the users and their processes in the information system. An object is a container of the information system. The set of objects includes active entities and passive entities of the information system. The subjects can access the objects by using actions like "read" or "write". DAC models use a matrix *A* to specify the permissions which are relations between the subjects, objects and actions. For example, *A(s, o)* defines the set of actions *a* that a subject *s* is authorized to perform on an object *o*. In the model HRU, the security rules are triplets <s, o, α> and these sets of triplets constitute the permission matrix. These models present several problems: - The administration of the permission matrix; - the super-user possesses all the right and his actions are not controlled by the security policies.

B. Roles Based Access Control models (RBAC)

The RBAC [21, 10] facilitate the security administration by associating the permissions to the roles and not directly to the subjects anymore. The two relations of Figure1 «*holds (Role, Permission)* » and « *Play (Subject, Role)* » define precisely the permission granted to each subject. A role can have several permissions and a permission can be associated to several roles. Equally, a subject can play several roles and, a role can be executed by several subjects.

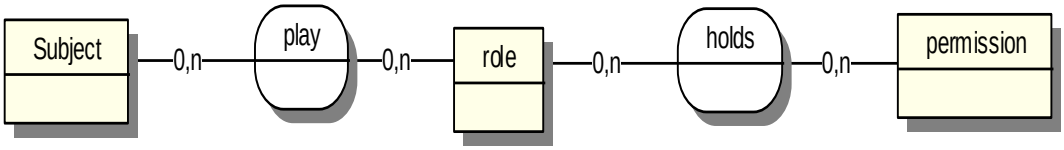

Fig.1: The Model RBAC

Some variants of RBAC models include the concepts of session and role hierarchy [26, 25, 12, 11, 2]. This permits a subject to activate in a session the roles necessary for the realization of the task to do. The concept of role hierarchy enables –Eiter to put in place a mechanism of inheritance of permissions between the roles and simplifies the administration of this model.

These models present several disadvantages: - the concept of permission is primitive and rigid since their usage and structure are dependent on the concrete application of the model; – the hierarchy of the roles ambiguous and do not correspond to the organizational hierarchy. For example, a bank director has a higher administrative role as compared to the accountant's. But, a bank director is not necessarily an accountant; – the model RBAC does not specify the permission which depends on the context. More precisely, if a permission is granted to a role, then all users who play that role inherit it; – the RBAC models do not allow the definition of a policy in a decentralized organization; - like the other models, they grant absolute rights to super-users and do not propose any control mechanism of his activities.

III. ORGANIZATION ROLES BASED ACCESS CONTROL MODELS (Or-BAC)

Or-BAC model [14, 22] focuses on the organization. It uses the entity-relation model and the first order logic to propose an expression language of the security rules. An organization can be seen as a group of structured active entities playing various roles. The concepts of this model are the following:

A. Subjects and Roles

A *Subject* can either be an active entity like a user or a passive entity like an organization. The entity role is used to structure the link between the subjects and the organizations. If *org* is an organization, *s* an object and *r* a role, then *Authorize (org, s, r) (Figure2)* means that org authorizes the subject *s* to play the role *r*.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (203-217), Month: July - September 2016, Available at: www.researchpublish.com

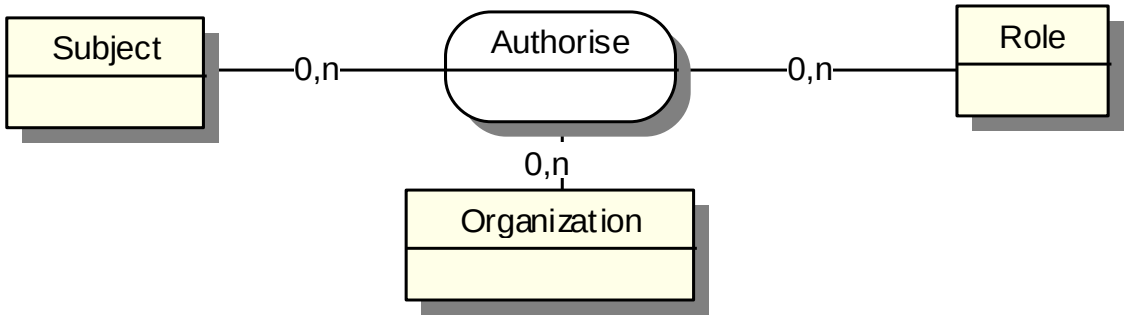

Fig.2: The relation Authorise

B. Objects and Views

The *Object* entity represents non passive entities. This model uses *views* to structure the objects. The views correspond to a set of objects which satisfies a common property. If *org* is an organization, *o* an object and *v* a view, then the relation *Uses (org, o, v)* (Figure3). Signifies that *org* uses the object *o* in the view *v*.

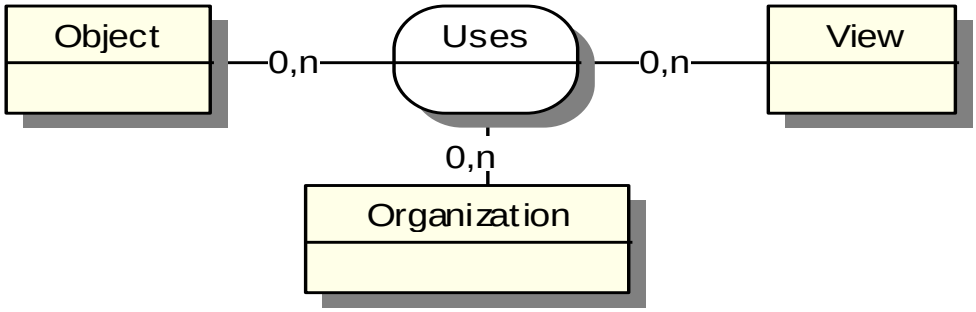

Fig.3: The relation *Uses*

C. Action and Activities

The *Action* entity embodies mainly basic computer operations like *"read", "write"*. The *Activity* entity is used as abstractions of actions: the activities corresponding to actions which have a common objective such as "*consulted*", "*modified*". If *org* is a organization, *α* is an action and a is an activity, then *Considers (org, α,* a*)* (Figure4) means that the organization *org* considers the action α as part of the activity a.

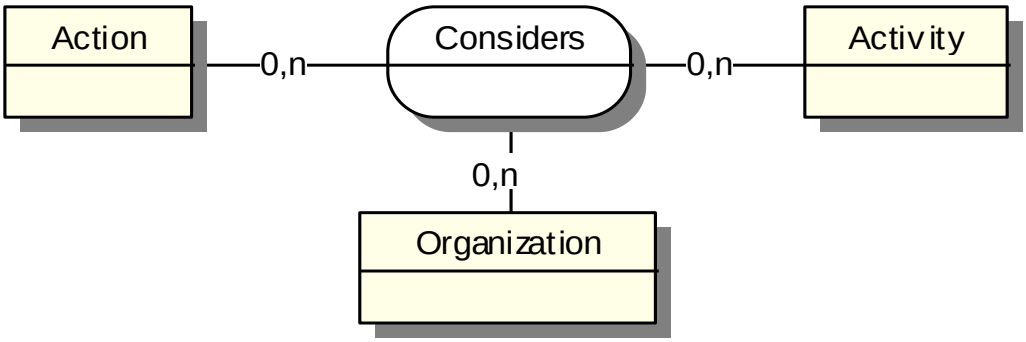

Fig.4: The relation considers

D. Contexts

The contexts are use to specify the concrete circumstances in which organizations grant permissions to perform activities on views. The contexts can be seen as relations between the subjects, the objects and the actions defined in a particular organization. If *org* is an organization, *s*, an object; α, an action, *o*, an object and *c*, a context, then the relation *Defines(org, s, α, o, c)* (Figure5) means that within the organization *org*, the context *c* is true between the subject *s*, the object *o* and the action α.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (203-217), Month: July - September 2016, Available at: www.researchpublish.com

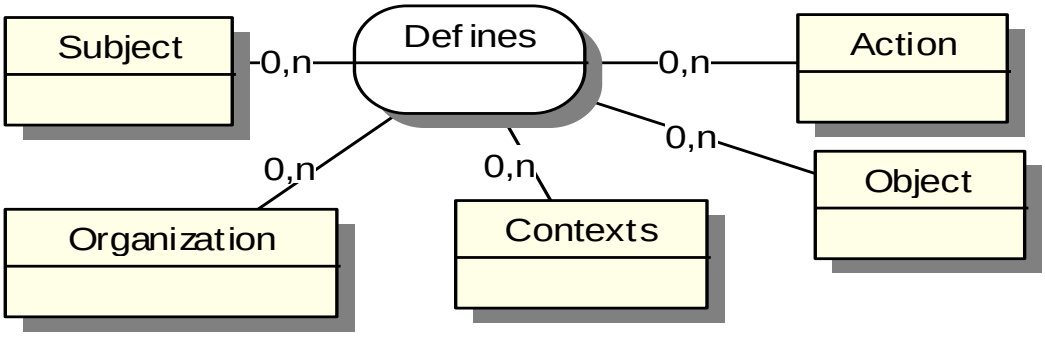

Fig.5: The relation *Defines*

The model Or-BAC proposes a language based on the first logic order to specify the conditions required for a given context to be linked, in a particular organization, to the subjects, objects and actions.

E. Security policies Or-BAC

The permission is a relation between organizations, roles, views, activities and contexts. The relations Interdiction, Obligation and Recommendation are defined in the same way (Figure6). If *org* is an organization, *r* a role, *a* an activity, *v* a view and *c* a context, then *Permission (org, r, a, v, c)* means that the organization *org* gives to the role *r* the permission to realize the activity *a* on the view *v* in the context *c*.

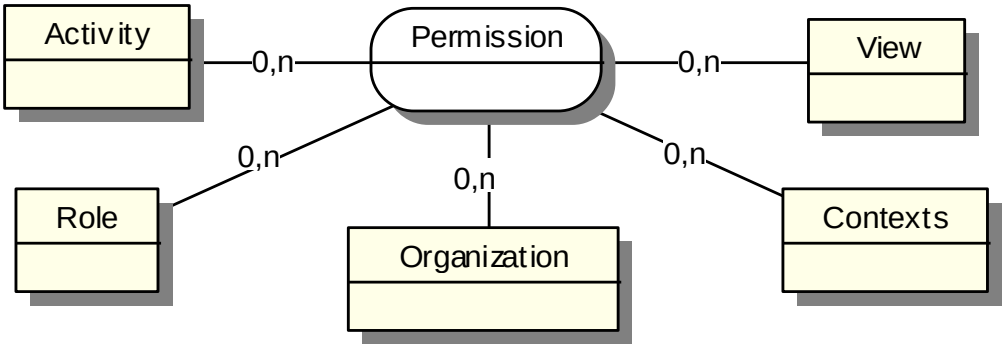

Fig.6: The relation *Permission*

The concrete authorization permits to write the concrete actions which cause the subjects to impact the objects Or-BAC introduces the relation *is_allowed* between the subjects, objects and actions: if *s* is a subject, α an action and *o* an object, then the relation that is allowed *(s, α, o)* means that the subject *s* has the permission to realise the action α on the object *o*. The relations: Forbidden, Is_obligatory and Recommended are defined in the same way. The model Or-BAC has the advantage of specifying generic permissions, thereby facilitating the administration of this model. It is used in decentralized organizations. It specifies the interdictions, recommendations and obligations in a well defined context.

However, this model has some drawbacks: - like its predecessors who does not propose any mechanism to control the super-user and grants him absolute powers on system resources; - it does not prevent the addition of fictive entities in the information system ; - the concept of hierarchy is defined like an inheritance of authorizations among the roles.

IV. ORGANIZATIONAL AND HIERARCHY BASED ACCESS CONTROL (HO-BAC)

In this section, we present our model HOr-BAC using the concept of organizational structure diagrams of the model entities-association (EA) then a formal language based on the first logic order. The concepts of organization, context and views are defined in the OR-BAC model. Arcs

A. Entities

It represents the enterprise in a hierarchy tree [20,9], where the internal nodes are administrative units, the leaves are operational units and the arc represent the information circulating between the nodes (Figure7). The organizational hierarchy is specified by the properties on the hierarchy tree.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (203-217), Month: July - September 2016, Available at: www.researchpublish.com

1) Administrative unit:

This entity Stands for the administrative roles Whose function is responsible for the control, supervision and validation of the requests emitted by the operational units. The Figure7 shows an example of an organizational tree where the node ($UA_{n,1}$) is the root and the intermediary node are administrative units ($UA_{n-1,2}$. $UA_{n-1,2}$. $UA_{n-1,2}$. $UA_{n-1,2}$.)

2) Operational Unit

This entity regroups the work the employees who have got the same training, roles and a specific function in an organization. It is found at the level of the leaves of the tree which represent the structure of the organization ($UO_{1,k}$;$UO_{1,k'}$)(Figure7). An operational unit does not take any decision on the functioning of the organization. It executes orders coming from the hierarchy under which it is placed, and emit a request to the hierarchy.

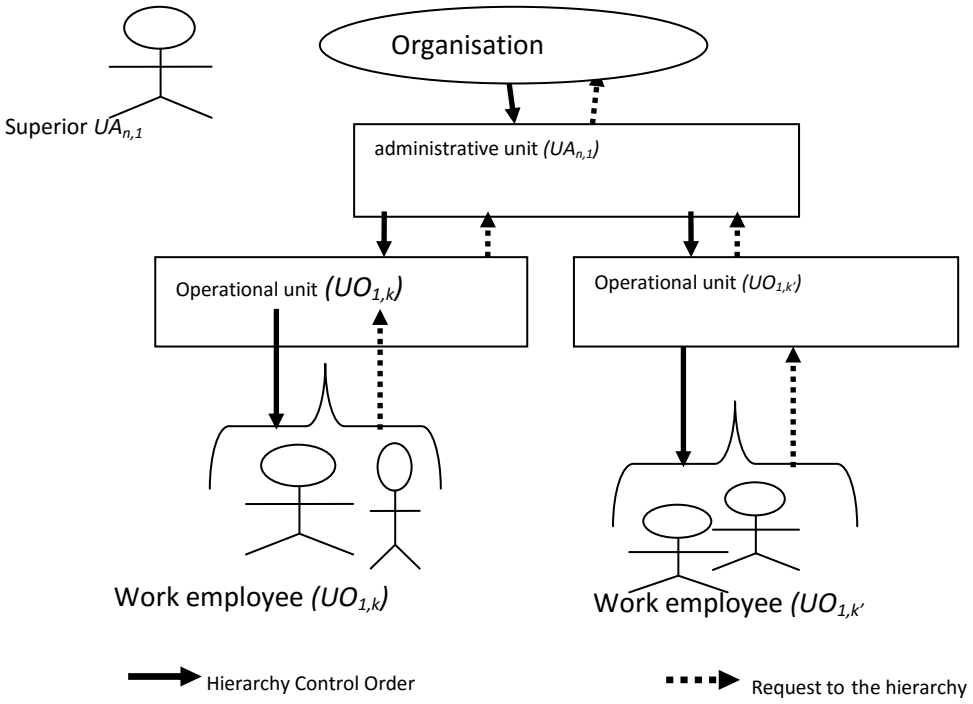

Fig.7: Organisational Structure.

3) Work employee

This entity is used to only represent active and human entities of the system. This permits to limit the users of the system to the employees organization.

4) Resources

The concept of resources is used to express the passive entities of the organisation. The concept of object is ambiguous in a case where it confuses a user and the resources he manipulates.

5) Requests

This entity enables to structure the request of change of state of resources by the operational units. A request can be a demand of «*consulted*», or of «*delete*». The request has the advantage to allow the administrative units to validate the activities of the personnel employed at the operational units it controls, before their effective execution on system resources.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (203-217), Month: July - September 2016, Available at: www.researchpublish.com

B. Association

1) Employee and operational unit

In an organization, a employee is transferred to an operational unit of the organization: If org is an organization, e, a employee and *uo*, an operational unit, then the relation *Employs (org, e, uo)* (Figure8). means that org Employs the employee e at the operational unit *uo*

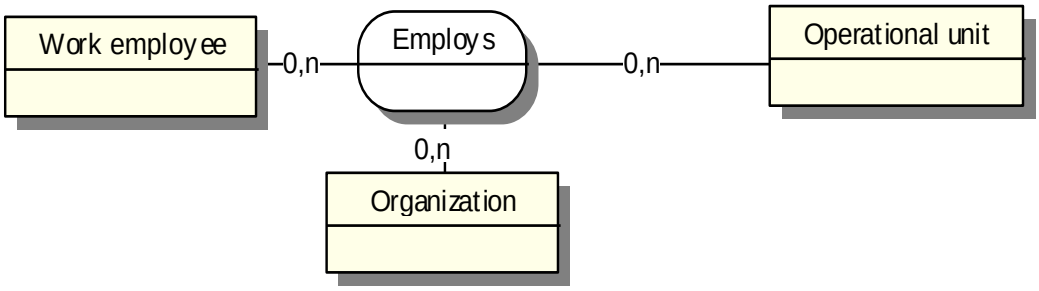

Fig.8: The Relation *Employs*

2) Work employee and Administrative unit.

In an organization each administrative unit is managed by a work employee: If *org* is an organization, *e* a work employee and *ua* an administrative unit, then the relation *Appoints* (*org*, *e*, *ua*) (Figure9). means that *org Appoints* the employee *e* at the head of an administrative unit *ua*

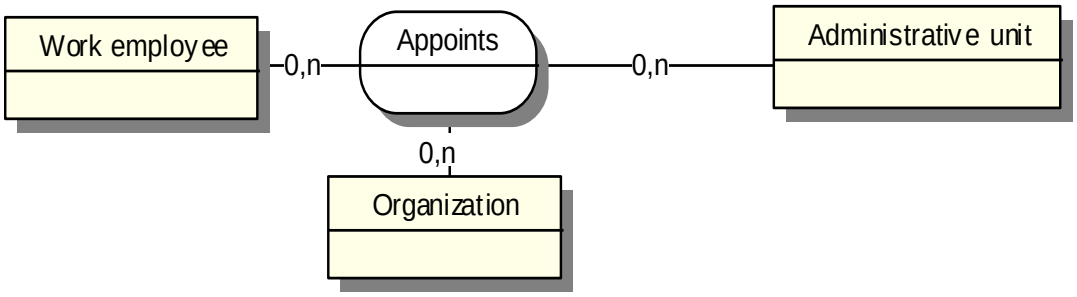

Fig.9: The Relation Appoints

3) Administrative unit and Operational unit.

In the Organisation, operational roles are placed under the administrative roles, because the operational unit has an executive functions, while administrative unit has control and supervise functions. If *org* is an organisation, *ua* an administrative unit, *uo* an operational unit then, the relation *place under (org, uo, ua)* (Figure10) implies that *org* places an operational unit *uo* under an administrative unit *ua*

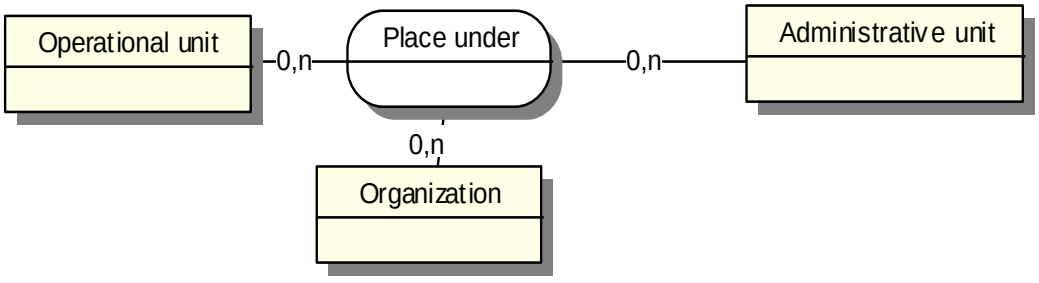

Fig.10: The Relation *Place under*

4) Administrative unit and Administrative unit.

In the Organisations hierarchy, apart from the administration council which is a special unit, all the administrative units are prioritized and each administrative unit is subordinate to another which insures the control of its activities. If *org* is an organization, ua_1 and ua_2 two administrative units, then the relation *subordinate* (*org*, ua_1, ua_2) (Figure11) signifies that *org subordinate* the administrative unit ua_1 to administrative unit ua_2

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (203-217), Month: July - September 2016, Available at: www.researchpublish.com

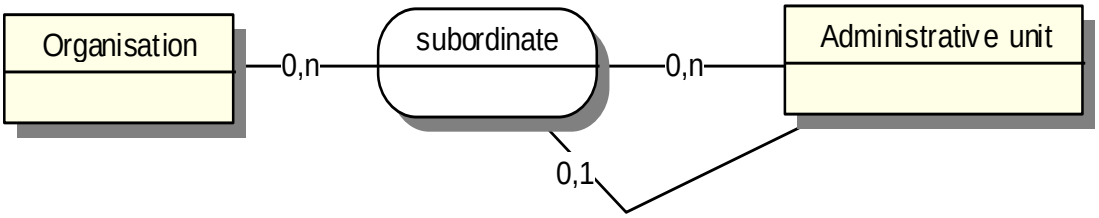

Fig.11: The Relation *subordinate*

C. Treatment Mode of a request

We introduce the concept of treatment mode to specify the request with *immediate* execution and request with *differed* one. The differed request goes through the hierarchical treatment and *immediate* request is executed without waiting for the hierarchical control.

D. Security policies

The relations Administrative permission and Operational permission correspond to those among organizations, operational and administrative units, views, requests and contexts and treatment mode. If org is an organization, *uo*, an operational unit, *ua* an administrative unit, q, a requests, v, a view, *c*, a context and m, a treatment mode, then the relation *Operational permission(org, uo, q, v, c, m)* (Figure12) means that the organization org grants to the operational unit *uo* the permission to incite the request q on the view v in the c contexts treated in m mode and the relation *Administrative permission (org, ua, q, uo, v, c, m)* (Figure13) means that the organization org grants the administrative unit *ua* the permission to process the request *q* emitted by the operational unit *uo* on the view *v* in the context c, treated in m mode

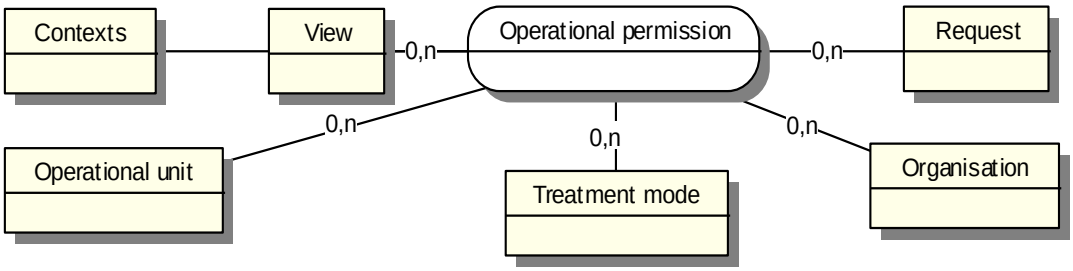

Fig.12 : The operational permission relation

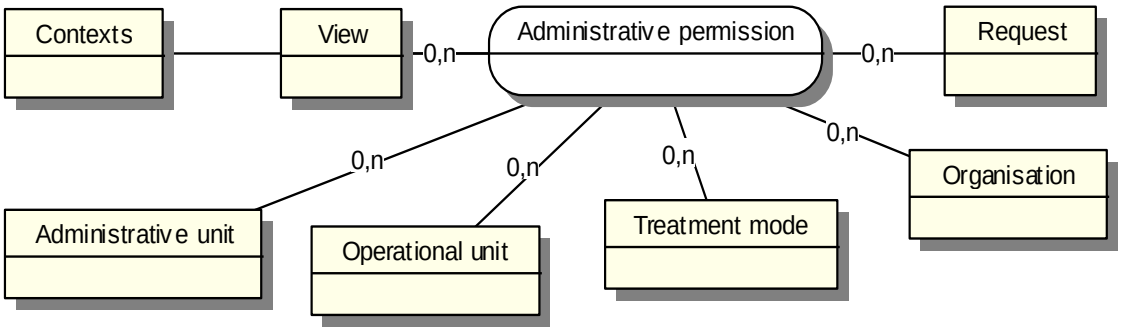

Fig.13: The administrative permission Relation

E. Concrete authorizations

With the aim of framing the concrete permissions of emissions of request, we introduced in this model the relation *can suggest* between the *employees*, *resources*, *actions* and *treatment mode*: If *e* is an employee, *α* an action, *r* a resource and *m* a treatment mode, then the relation *can suggest (e, α, r, m)* meaning that the employee *e* has the permission to suggest the application of the action *α* on the resource *r* in the *m* treatment mode. Since this relation enablesonly one suggestion we will equally introduce another permission which defines the concrete expression of treatment of a suggestion. It is the

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (203-217), Month: July - September 2016, Available at: www.researchpublish.com

relation *can treat* between the subordinate employees and hierarchically superior employees, the resources, actions and *treatment mode*. If *e* is a subordinate employee, *e'* a hierarchical superior employee of *e*, *α* an action; *r* a resource and *m* a treatment mode, then *can treat (e', e, α, r, m)* means that the employee *e'* has the permission to treat the suggestion of *e* on the application of the action *α* on the resource *r* in the *m* treatment mode.

F. The model Entity-Association

The Figure14 sums up our security model. It contains nine entities (Organisation, employee, operational unit, administrative unit, Resource, View, Action, Activity, Context and Treatment Mode) and eleven relations (Appoints, Employs, Placed under, Subordinate, Uses, Consider, Operational Permission, Administrative Permission, Can suggest, Can treat and Define).

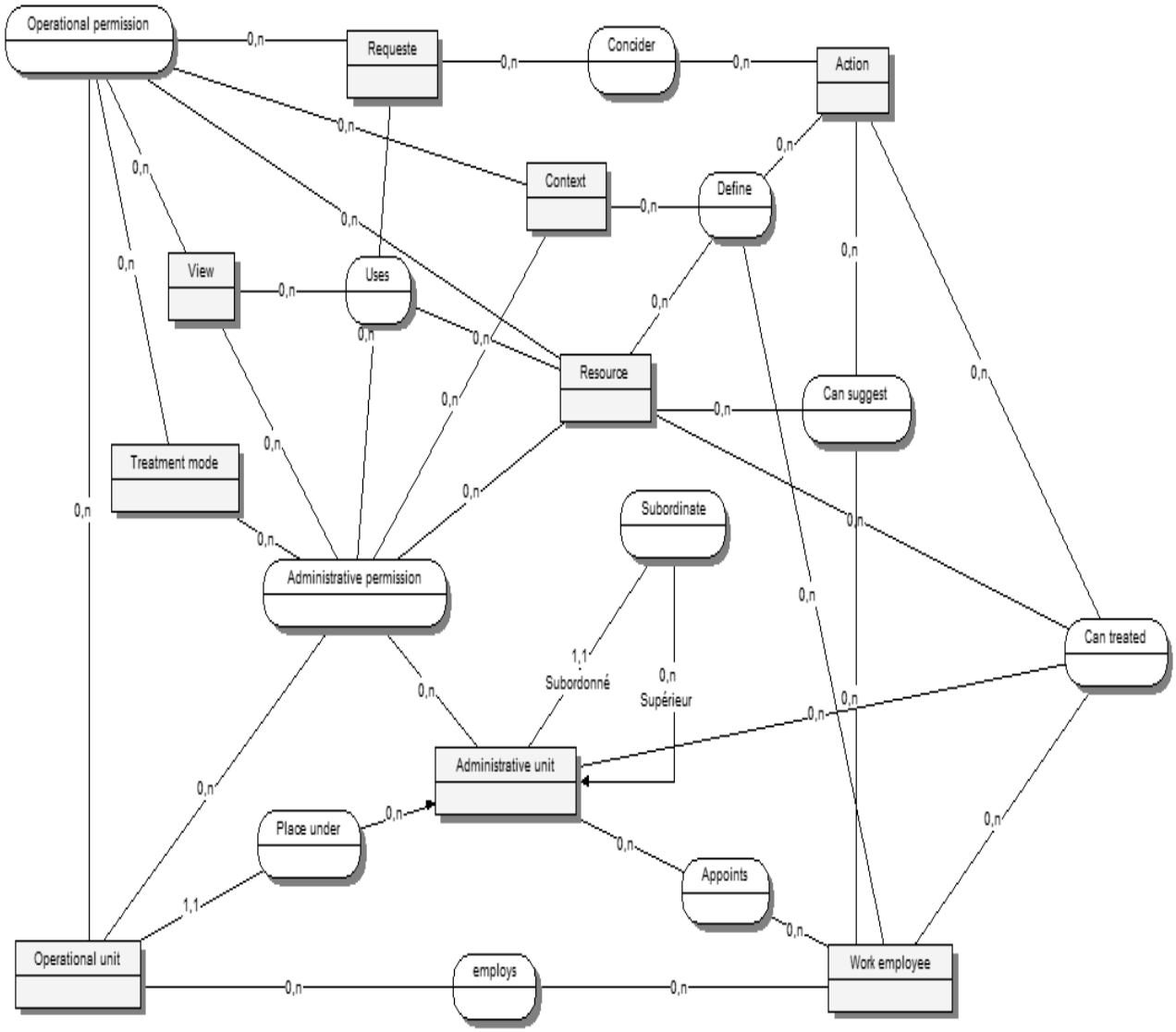

Fig.14: The HOr-BAC Model

V. LANGUAGE PROPOSED FOR HOR-BAC

We propose in this section language «*L*» based on a first logic order, derived from the language proposed for Or-BAC[15, 14]. Each expression of *L* will contain symbols extracted from a particular vocabulary classified in four groups: *constant symbols, individual variables, symbols of relations and symbols of functions.*

A. Constants

It corresponds to the models entity EA they are denoted by lower case letters like *a*, *b* and *c*. The types of constant's symbol are entities of the HOr-BAC model.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (203-217), Month: July - September 2016, Available at: www.researchpublish.com

B. Variables

Variables are denoted by lower case letters like *x*, *y* and *z*. There are individual variables for each type θ. The constant symbol of type θ and the individual variables of type θ will be called *terms-*θ.

C. Relation

The symbols of relation *L*, denoted by words starting with upper case letters, corresponding to the relations of our diagram EA. Each symbol of relation of *L* is considered as a type of relation.

D. Functions

To extract information from sum entities properties we use *functions*. The symbols of functions are denoted by lower case letters like *f, g and h*. For example, to derive the attribution *Name* from the employee entity we use a symbol of function corresponding to the attribution *Name* of domain *Employee* type and of co-domain the set of names. In order to derive the information represented by the symbols of function, we have to introduce concrete binary relations among the domains.

E. Atomic formulas

Properties and actions define the fundamental elements of the language. Using the terms and the relations, we make up the atomic formulas as follows: if t_1 is a term- θ_1, …, t_n is a term-θ_n and *P* is a relation of type (θ_1, … θ_n), then $P(t_1, t_n)$ is an atomic formula. For example : *Employee* (UDS, Félix, teaches), *operational permission* (UDS, Paul, consult, notes, teaching) and administrative *permission* (UDS, Jules, consult, department, notes, teaching), *operational permission* (UDS, Célestin, consult, notes, jury) et *administrative permission* (UDS, Rudoph, consult, department, notes, jury) are atomic formulas.

1) Semantics:

The formulas will considered as atomic formulas. This way the formulas of L are defines as follows : An atomic formula is a formula ; - if A is a formula then; A «not A» is a formula ; if A and B are formulas then (A ∧ B) «A and B» and (A ∨ B) «A or B» are formulas if A is a formula and x is an individual variable then ∀xA «for all possible variables of x, we have A» et ∃xA «there exist possible values of x such that A» are formulas. The logic connectors → and ↔ are defined in the following way: (A → B) is equivalent to (¬A ∨ B), and (A ↔ B) is equivalent to ((A → B) ∧ (B → A)). We will voluntarily omit the brackets when no ambiguity is possible. This is an example of a formula ∀e (Employment (UDS, e, lecturer) → Employee (UDS, e, teacher)): this means that «all those lecturers in the university of Dschang are also teachers».

2) The truth conditions

We suppose that the security policies will be based on the following list of axioms proper to all the organisations.

∀e∀α∀r∀uo∀q∀v∀c∀m operational Permission(org, uo, q, v, c, m) ∧ Employs(org, e, uo) ∧ Uses(org, r, v) ∧Consider(org, α, q) ∧Define(org, e, α, r, c) → can suggest(e, α, r, m) *: which means that «if the organization* org, *in the context* c, *grants the operational permission to the operational unit uo to emit the request q on the view* v *and if org Employs the employee* e *to at the operational unit uo and if* org *uses the resource* r *in the view* v *and if* org *considers that the action* α *can be used to emit the request* q *and if within the organization org the context* c *is true between* e, α , r, *then the employee* e can suggest *the application of the action* α *on the resource* r *and this suggestion is treated in* m *treatment mode »* ,

∀e∀α∀r∀uo∀ua∀q∀v∀c∀m Administrative permission (org, ua, q, uo, v, c, m) ∧ Superior (org, e, Emitter(q)) ∧ Uses(org, r, v) ∧Consider(org, α, q) ∧Define(org, e', α, r, c) → Can treat (e, Emitter(q), α, r, m) *: which means that «if the organization* org, *in the context* c, *grants the administrative permission to the administrative unit* ua *to treat the request* q *coming from the organizational unit* uo *on the view* v *in mode* m *and if in* org e *is the superior of the request emitter* q *and if* org *uses the resources* r *in the view* v *and if* org *considers that the action* α *can be used to emit the request* q *and if within the organization* org *the context* c *is true between* e', α *and* r, *then the employee* e *can treat the request application of the action* α *on the resource* r *emitted by the emitter of the request* q, *in* m *treatment mode*».

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (203-217), Month: July - September 2016, Available at: www.researchpublish.com

F. Hierarchical order in an organization

Several security models have defined the notion of role hierarchy in an organisation [63]. Their interpretation of the order between the roles brings about many contradictions. Other models [85] have added *bans*, *recommendations* and *obligations* to solve these contradictions. This creates situations where an action is permitted and forbidden to the same subject. We propose a partial hierarchical order having on the one hand the employees and on the other hand the organizational units (Figure2).

Hierarchical order between organisational units: *Relation Hierarchy superior between two organizational units'* uo_1 *and,* uo_2 *of an organization org* specified *in the following way:* Hierarchy superior (org, uo_2, uo_1)→Is administrative unit(uo_2)∧ ((Is administrative unit(uo_1)∧ Subordinate(org, uo_1, uo_2)) ∨ (Is operational unit(uo_1)∧Place under(org, uo_1,uo_2))). *This relation means that* uo_2 *is the hierarchical superior of uo1 (*uo_2> uo_1*) in the organization org if* uo_2 *is an administrative unit and that* uo_1 *is an administrative unit that org Subordinates at* uo_2 *or that* uo_1 *is an operational unit that org Places under* uo_2.

Hierarchical order between the employees. Relation superior between two employees' e_1 and e_2 of an organization *org* specify by the following way: Superior (org, e_2, e_1)→They_work_in_an_administrative_unit(e_2)∧ (They_work_in_an_administrative_unit(e_1)∧ Subordinate(org, Unit(e_1) , Unit(e_2)) ∨ (They_work_in_an_operational_unit(e1)∧Place under(org, Unit(e_1), Unit(e_2)))). This relation means that e_2 is the hierarchical superior of e_1 (e_2 >e_1) in the organization org if e_1 works in an administrative unit subordinate to an administrative unit managed by e_2, or if e_1 works in an operational unit placed under the administrative unit managed by e_2.

G. Algorithm electronic book signature

Is a procedure to secure automatic treatment based on the model HOr-BAC. It supposes that the totality of the actions in an organization are carried out on request, and each demand obtains a validation for an effective execution in the system if not the demand is rejected. This procedure is exposed in three phases. The phase of system initialization, the phase of emission and the phase of treatment of requests.

1) Initialization

The following procedure helps initialise the electronic book signature

Creat_org: creates the organizational structure in a tree and turns the root which is the largest administrative unit of the organization

Creat_emp ; Takes the list of personnel and transfers everyone to an organizational unit by using the relation employ and transfer.

create_view ; takes the list of resources and creates different views of the organization by using the relation uses

create_requests ; Takes the list of actions and creates the different requests of the organization through the relation Consider

create_treatment_mode ; creates the different treatment modes used in the organization.

create_permissions ; associates to each organizational unit the operational permissions or the administrative permissions following the various cases, and gives operational permissions to the work employees and assign their administrators.

2) *Emission*.

This algorithm controls the request emissions, takes into consideration the employee e, the action α, the resource r, the request q, the view v, the context c and the action mode m

Emission (e : employee, a : action, r : resource, q : request, v : view, c : context, m : treatment mode)
Begin
If(Operational permission (org, Ammeter(q), q, v, c, m) and
Employs (org, e, Ammeter(q)) and
Uses (org, r, v) and
Consider (org, α, q) and

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (203-217), Month: July - September 2016, Available at: www.researchpublish.com

Define (org, e, α, r, c)
then
If m=immediate then
Execute (a, q) ; {Applied action(α) on resource(r)}
Saved result() ;
Else {created a request}
Can suggested (e, α, r, m) ;
Saved request () ;
Run Alert request available();
Endif
Endif
End.

3) *Request processing.*

The algorithm of treatment takes into consideration the employee e, the action α, the resource r, the request q, the view v, the context c, and the action mode m

Processing (e : employee, a : action, r : resource, q : request, v : view, c : context, m : treatment mode)
begin
If (Administrative permission (org, Unit(e), q, emitter(q), v, c, m) and
Superior (org, e, emitter(q)) and
Uses (org, r, v) and
Consider (org, α, q) and
Define (org, e, α, r, c)
then
If Height(Unit(Recipient(q)))=Depth(Emitter(q)) then;
{final treatment: accord or refuse of the hierarchy }
Can treated(e, emitter(q), α, r, m)
Saved result() ;
Else { intermediaries treatment }
Can treated(e, emitter(q), α, r, m) ;
Saved request() ;
Run Alert request available();{transmission}
Endif
endif
end

VI. SUPER-USER CONTROL EXAMPLE

This section focuses on the way to express a simple example of security policy to control the super-user activities.

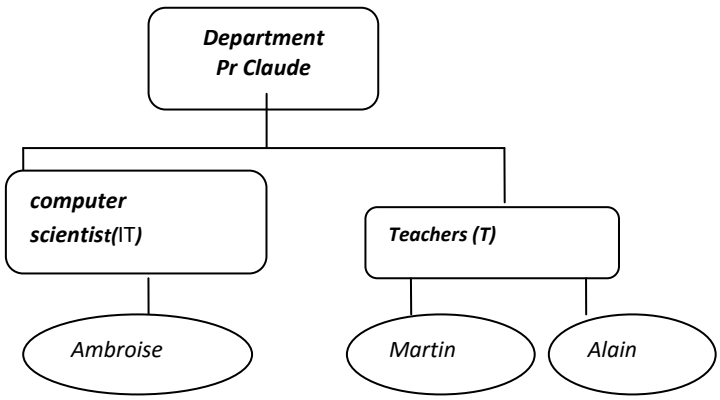

Fig.15: Organizational Structure f an academic department

A. Employees and organizational units

In this organization the operational units are Computer Scientist (CS), Teacher (E) an administrative unit, the department.

B. Employees and operational units

The organization of the Figure 14 Uses Ambrose for the operational unit IT, Alain and Martin in the operational unit Teacher. To represent this fact we used instances of the relationship Employs:

Employs (UDS, Ambrose, IT)

Employs (UDS, Alain, Teacher)

Employs (UDS, Martin, Teacher)

C. Employees and administrative units

The organization of the Figure 14 appoints Prof. Claude at the administrative unit *Department*. To bear this fact out we used the instance of the relationship *appoints (UDS, Claude, Department).*

D. resources and views

To simplify this example we considered the department's resources in thou views:

Administrative

administrative information on all the resources of the department. eg the file SalaireTBL is in this view and this fact is expressed by the instances of Uses relationship as follows: Uses (UDS, SalaireTBL, Administrative)

Technique:

Technical information on all the resources of the department. eg the file SalaireTBL is in this view and this fact is expressed by the instances of Uses relationship as follows: Uses (UDS, SalaireTBL, Technique)

E. Actions and Request

The organization considered three requests: *Creation* to *insert*, *modification* to *update*, *consultation* to *select*. This is specified by the following facts: *consider*(UDS, *insert*, *creation*),

F. Treatment modes

This organisation uses two treatment modes: *immediate*. The request is implemented immediately without hierarchical treatment; *Normal*. the request must be treated by the head of department before its execution in the system

G. Contexts

The *maintenance* context that binds the super-user *Ambrose* to the change request in the SalaireTBL table of system view is defined by: *Définit (UDS, Ambrose, update, SalaireTBL, technique)*

The administration context that binds the head of department Claude to the change requested in the SalaireTBL table of Administrative view is defined by : *Définite (UDS, Claude, update,* SalaireTBL, *Administration)*

H. The hierarchy

To specify that the operational units are under the administrative unit department, we use the following rule.

Place_under (UDS, computer scientist, Department).

Place_ under (UDS, Teachers, Department).

I. Specification of safety rules

To specify the following operational permission: a super-user can immediately consult the file of the system view in a technical context. But to change he must wait the decision From the head of department. We use the following rule:

operationnal_permission (UDS, computer scientist, consult, system, technique, immediat)

operationnal_permission(UDS, computer scientist, modification, system, technique, normal)

And the concrete operational permissions are derived as follows:

∀e operationnal_permission (UDS, computer scientist, consulter, System, technique, immediate)∧ Employs(UDS, e, computer scientist)∧Uses(UDS, SalaireTBL, system)∧Consider(UDS, select, consult)∧definie(UDS, e, select, SalaireTBL)→Peut_suggest(e, select, SalaireTBL, immediate)

∀e operationnal_permission (UDS, computer scientist, modification, system, technique, immediate)∧ Employs(UDS, e, computer scientist)∧Uses(UDS, SalaireTBL, system)∧Consider(UDS, select, modification)∧ definie (UDS, e, updat, SalaireTBL)→Peut_suggest(e, update, SalaireTBL, immediate)

permission for the head of department to process requests of modification we use the following rule:

Administrative_permission (UDS, department, modification, system, administration, normal)

And the concrete operational permissions are derived as follows:

∀e Administrative_permission (UDS, department, modification, child(department), system, administration, normal)∧ appointed (UDS, Claude, department) ∧(computer scientist ∈ descending (department)) ∧Uses(UDS, SalaireTBL, system)∧Consider(UDS, update, modification)∧definie(UDS, Claude, update, SalaireTBL, normal)→Can_treated(Claude, update, SalaireTBL, normal)

J. Electronic signature book

We assume that the signature book is already initialized through created_org; created_emp; created_view; created_request; created_mode; created_permission functions. Suppose Ambrose superuser issues the following salary modification request. UPDATE salairetbl SET index = 645 serial WHERE = 'EQ54W45';

Requests emission:

The control algorithm emission (Ambrose, UPDATE, salairetbl, change, system, technique, normal) intercepts the request and performs control. At the end of this control, a request available alert is sent to the head of department.

Request treatment:

In the attribute hierarchical decision, to request the head of department approval or reject and the algorithm *Traitement(Claude, update,* salairetbl*, modification, system, administration, normal) run. If the head of depatment valid the request* If the head of department has validated the request while the index of the serial employee 'EQ54W45' becomes 645 else the index does not change. Note that the head of department can be informed by SMS and process the request through the same channel.

VII. CONCLUSION

Our concern in this paper was to propose a new security concept, the electronic signature book, based on an extension of Or-BAC model that we have developed and called HOr-BAC. This concept of electronic signature book permits to control a request emission, treatment and execution of the super-user. The request here represents the employees' activities on the information system.

Our HOr-BAC model includes the following concepts: The concept of *organisational unit*, has enabled the new model to take in consideration the real hierarchical exigency of the organisations in the treatment of data; The concept of *request* has helped frame the procedure of data treatment via different organizational units and allowed the control and the human and hierarchical validation of the activities of all subordinate employees. By so doing, the activities of the super-users are controlled and validated by the hierarchy; the concept of *treatment.* The « *treatment mode* » has permitted us to mark those actions which demand a strict control in order to submit them to the hierarchical validation.

The approach has been used to secure a bank agency information system on the SGBD PostgresSQL using 10 employees and several applications. On a six months exploitation duration no internal attack has been detected.

Several problems have not been investigated in this article. We have not discussed administrative problems of the security policy. Obviously, an extension of the administration model RBAC presented with ARBAC [27] is being developed. In the same vein, the definition of a security policy by the model HOr-BAC will be looked Through in a subsequent article.

REFERENCES

[1] M. A. ABAKAR « Etude et mise en œuvre d'une architecture pour l'authentification et la gestion de documents numériques certifiés Application dans le contexte des services en ligne pour le grand public ». Thèse soutenue publiquement à l'université Jean Monnet de Saint-Etienne le 22 novembre 2012.

[2] G. Ahn and R. Sandhu, "Role-Based Authorization Constraints Specification", ACM Transactions on Information and System Security (TISSEC), vol. 3, n° 4, novembre 2000, pp. 207-226.

[3] D. E. Bell, L.J. LaPadula, Secure Computer Systems : Unified Exposition and Multics Interpretation, Rapport technique, MTR 2997 Rev. 1, MITRE corp., Bedford (Massachussetts, USA), 1976.

[4] E. Bertino, P. A. Bonatti, et E. Ferrari, « TRBAC: A temporal role-based access control model », ACM Trans. Inf. Syst. Secur. 4(3): 191-233, 2001.

[5] K.J.Biba, Integrity Consideration for Secure Computer Systems, The MITRE Corporation, Technical Report ESD-TR-76-372 & MTR-3153, 1977.

[6] CSI(*Computer Security Institute*) /FBI Federal Bureau of Investigation),Sondage: cyber-anarchie ou cyber-mensonge made in USA ? Mercredi 10 avril 2002 *http://solutions.journaldunet.com/0204/020410_cybercrime.shtml*, accédé le 20/08/2006

[7] F. Cuppens et A. Miège ENST Bretagne, Oganization Bassed Access Control Article 27 mai 2004 Campus de Rennes 2, Rue de la Chataîgneraie 35576 Sesson Sévigné CEDEX.

[8] F. Cuppens, N. Cuppens-Boulahia, et C. Coma-Brebel, « MotOrBAC : un outil d'administration et de simulation de politiques de sécurité ». First joint conference on security in network architectures (SAR) and security of information systems (SSI), 6 -9 June, Seignosse, Landes, France, 2006.

[9] H. Fayol, Administration industrielle et générale, Parie, Edition dunod, 1979.

[10] D.F. Ferraiolo, R. Kuhn, R. Sandhu, « RBAC Standard Rationale: comments on a Critique of the ANSI Standard on Role Based Access Control, » Dans IEEE Security & Privacy 2007, vol. 5, no. 6, pp. 51-53, 2007.

[11] D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn and R. Chandramouli "A Proposed Standard for Role-Based Access Control", ACM Transactions on Information and System Security, vol. 4, n° 3, août 2001.

[12] S.I. Gavrila, J.F. Barkley, "Formal Specification for Role Based Access Control", Third ACM Workshop on RBAC, Fairfax, Virginia, USA, 22-23 octobre 1998.

[13] M. A. Harrison, W. L. Ruzzo et J. D. Ullman. Protection in Operating Systems. Communication of the ACM, 19(8):461-471, Août 1976.

[14] A. A. E. Kalam, Rania El Baida, Philippe Balbiani, Salem Benferhat, Frédéric Cuppens, Yves Deswarte, Alexandre Miège, Claire Saurel, Gilles Trouessin, CRIL, ENST, Ernst & Young Audit, IRIT, LAAS-CNRS, ONERA, un modèle de contrôle d'accès basé sur les organisations, Revue InfoSyst 2003

[15] A. A. E. Kalam, A. Marzouk, TOrBAC : A Trust Organization Based Access Control Model for Cloud Computing Systems, 2012.

[16] A. Kamoun. « Adaptation d'architectures logicielles de contrôle d'accès dans les environnements collaboratifs ubiquitaires. Réseaux et télécommunications » cs.NI]. Universite de toulouse 1, 2014. Français. <tel-01200743>

[17] B. Lampson, "Protection", 5th Princeton Symposium on Information Sciences and Systems, 1971

[18] M. Liu and X. Wang, "Security analysis on gtrbac model and its improvement", Computer Applications and Software, vol. 29, vol. 10, (2012)

[19] M. Liu, and Xuan Wang « Safeness Discussions on TRBAC and GTRBAC Model and an Improved Temporal Role-Based Access Control Model » International Journal of Security and Its Applications Vol.9, No.8 (2015)

[20] H. Mintzberg, Le management : Voyage au centre des organisations. Paris : Ed. d'Organisation 1998.

[21] Nationale Institute of standards and Technology (NIST), Site Web: http://crs.nist.gov/groups/SNS/rbac/, 2011.

[22] Le modèle de contrôle d'accès Or-BAC Site web : http://www.orbac.org/, 2013.

[23] M. B. Saidi, A. Marzouuk, Multi-Trust-OrBAC : Access Control Model for Multi-Organizational Critical Systems Migrated to the Cloud, 2013.

[24] R.S. Sandhu, "Expressive Power of the Schematic Protection Model", Journal of Computer Security, vol.1, n° 1, pp. 59-98, 1992.

[25] R.S. Sandhu, "Role Hierarchies and Constraints for Lattice-Bases Access Controls", in 4th European Symposium on Research in Computer Security (ESORICS 1996), (E. Bertino, H. Kurth, G. Martella, E. Montolivo, Eds.), Rome, Italie, september 25-27, Lecture Notes in Computer Science 1146, pp. 65-79, ISBN 3-540-61770-1, SpringerVerlag, 1996.

[26] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman, "Role-Based Access Control Models", IEEE Computer, vol. 29, n° 2, pp.38-47, février, 1996.

[27] R. Sandhu, Bhamidipati et Qamar Munawer. The ARBAC97 Model for Role-Based Administration of Roles. ACM Transactions on Information and System Security, 2(1), Février 1999.

[28] R. Thion, « Structuration Relationnelle des politiques de contrôle d'accès Représentation, Raisonnement et Vérification, » Thèse soutenue en 2008.

[29] *J. Zheng, k. Q. Zhang, w. S. Zheng, an Y. Tan* « Dynamic Role-Based Access Control Model » Journal of Software, Vol 6, No 6 (2011)