Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 8 (2013) pp. 067-075

XBACL: An Access Control Language for Financial Data

Ivanildo J. S. Aquino Júnior[1], Cleberton Carvalho Soares[2], Paulo Caetano da Silva[3], Sidney Viana[3], Valéria C. Times[1]

[1]Federal University of Pernambuco (UFPE)
Recife – PE – Brazil
{ijsaj,vct}@cin.ufpe.br

[2]Estácio College of Sergipe
Aracaju – SE – Brazil
cleberton.soares@estacio.br

[3]Salvador University - UNIFACS
Salvador – BA – Brazil
{paulo.caetano, sidney.viana}@pro.unifacs.br

Abstract: The fast growing of Internet research area has affected the way how enterprises have made their financial information available. In the same context, several security issues are becoming more important as time goes by because of the great value of such information. This paper presents XBACL (eXtensible Business Access Control Language), which is a language to specify access control policies for financial data. A XBACL formalism is particularly given, including its semantic and syntax. Most of the correlated work found in literature does not take into account the kinds of documents that have information about the information hierarchy found in other documents. XBACL implements this by using the linkbases hierarchy information. Results derived from the implementation of this language indicated to be more concise than the use of general purposes access control policies.

Keywords: XBRL; Access control policies; financial data; information security.

I. INTRODUCTION

In the global and interconnected economy, efficient and effective methodologies to communicate organizational financial information are increasingly important. Effective communication means organizing useful data for a variety of stakeholders, and storing that data in an accessible location and format to support regulation and shared analysis [Marshal et al, 2010]. Aiming to share information, 99% of the world's 500 largest companies have Websites and 94% of them include financial information. In this scenario, eXtensible Business Reporting Language (XBRL) [Engel et al, 2003] is the standard language for divulgating financial reports on the Internet. However, the Internet is an unsafe environment, for its own nature, and XBRL does not include mechanisms to ensure safety on the publication of these information, such as policies for access control, digital signature and encryption.

Among the security mechanisms mentioned above, the encryption and digital signature of XML documents [17, 22, 29, 30, 31] are solved issues, because through consultation of the specific literature we can find solutions that fit the needs of XBRL. However, with regard to access control certain deficiencies will be discussed below.

Many policies of access control to XML documents [2, 6] using the XML node as the smallest unit of protection and nodes to be accessed are specified by XPath language [8]. Moreover, the spread of a policy in an XML element is performed only in the XML document hierarchy. However, this approach does not seem to be the most appropriate to the context of XBRL.

The aim of this paper is to describe a model of access control policy of XBRL financial reporting, based on the definition of subreports, displaying the language *eXtensible Bussiness Access Control Language* (XBACL). This study has been developed in the XBRL *framework* project [32] at Salvador University. XBRL *framework* is a system for management and publishing of financial reports in XBRL format on the Internet. In this system, one of the most important components, in the security context, is the XBACL processor, responsible for managing policies of access control to the repository of reports.

The rest of the content presented in this paper is organized as follows. In section II, recommendations of the Worldwide Web Consortium (W3C) for security in XML documents are disposed. The section III discusses the main works related in the literature. The section IV describes the context of financial reports available on the Internet and XBRL language, including its most relevant features to this work. The section V contains the formalization of the concept of sub-reports used in the definition of XBACL. The semantics of the access control model is detailed in Section VI. In Section VII, we present a case study of the use of language XBACL including implementation details of an access

control module and, finally, section VIII contains the conclusion.

II. W3C Recommendations for XML Security Documents

The W3C is an international community focused on the web growth, through techniques specifications and recommendations or guidelines with relevant content for software developers and designers, beside companies, scholars and other interested players, especially those who belong to the Consortium. Among the technologies covered by it, XML is a standard developed and maintained by the W3C. Among the working groups at the W3C related research of XML technology, XML Security Working Group is the team that is responsible for the security activity, so it belongs to the group that standardizes the creation of safe environment for XML. This team has three groups of specific activities: (1) XML Signature, (2) XML Encryption, and (3) XML Key Management Specification - XKMS. Proposals contained in the specifications XML Encryption and XML Signature demonstrates the use of technology to spread information security, respectively: encryption and digital signatures. The XKMS - Key Management Specification is related to management infrastructure of public keys, acting in support of the other recommendations.

XML Signature defines syntax and processing rules for digital signature, dealing with the data to be signed as well as their representation in XML, and, in a opportune moment, the due inquiry signature, for example, when you attest the integrity of XML received.

XML Encryption operates on the scope of confidentiality, recommending the use of resource of cryptography to encrypt messages and represent it in the XML document. This process supports data from multiple sources: XML documents, elements or content of an XML element as well. Therefore, it is possible apply XML Encryption in an entire document or in part of it. It can also act in the reverse process of encryption, which is to make the content that was encrypted unreadable.

The XKMS specifies the service of public keys management in those features: registration, revocation and renewal; beside validation and location of public keys. Therefore, XKMS becomes an agent (third) for communication reliability, especially regarding the use of XML by companies, in the relation *business-to-business* (B2B).

The adherence of these specifications to the XML context supports the assumptions of confidentiality and integrity regarding to the protection of the information in a non-secure means of communication, such as the Internet.

However, we notice the absence of a third element of the triad of information security, named availability. You must identify the user who wants to access the information by an authentication process in order to avoid, for example, that unauthorized people have a not allowed or forbidden access to the XML document. For this purpose, the access control is indispensable because this is an assumption required to achieve a global proposal for the security of information in the scope of XML technology, and that is not yet defined by W3C.

III. Access Control to XML Documents

Among the current related works dealing with the subject, "Access Control to XML documents," Batista [2005] defines a formation of Policy Base, in which he deals with the generation of digital signatures by hash. The XBRL document now has mechanisms to audit the integrity and completeness of the document, something already much discussed in the literature. At this work, the XBRL document as treated, in which each part is defined as a node and each node is associated with a digital signature. It is based on the proposed model "Merkle Hash Tree" [13] and denominates the hash function that is used as "Merkle hash function", and XPath proposed as a mechanism of access control. However, we understand that the XPath has limitations, which we will describe later in this work.

BHATTI [2003] presents specification and architecture of the X-RBAC language, in order to support the implementation of a "Role-Based Access Control" (RBAC). The RBAC model is characterized by the notion that the permissions are assigned to roles and not directly to users. Users are assigned to appropriate roles according to their functions and thus indirectly acquire the permissions associated with these roles. The scheme is represented by the triad: user, rule and permission. Regarding the process of RBAC, this defines the rules of users, allow rules and signatures, generating and feeding the data according to the structure required by a taxonomy.

As for ROSSET [2004], from the need to act on restrictions of diffusion of DRM (Digital Right Management) using SAML (Security Assertion Markup Language) standard, which is used for exchanging messages between system entities acting as language rights distribution. The SAML proposal is based on the message exchange between the entities involved in the authorization process and distribution of rights. It is based on the architecture of authorization AAA Authorization Framework IETF [Vollbrecht, 2000, apud Rosset], and also in the conceptual model discussed in its own specification standard XACML [Godik, 2003, apud Rosset]. The entities that may be involved are: the PEP (Policy Enforcement Point), PDP (Policy Decision Point), PIP (Policy Information Point), PAP (Policy Administration Point) and PRP (Policy Retrieval Point). Most of these entities have the same features presented in standard XACML [Godik, 2003, apud Rosset].

GOWADIA [2003] presents as RDF statements may be used to apply access control in XML trees and its association, representing security objects and expressing a flexible and granular security policy. He sets the tripod (s, o, ±) to implement and enforce the rules of access control of the process, where: “s” is the subject, “o” is the due protection to the object; and ± the access mode. It presents the specifications of a language called RXALC - RDF-based Access Control Language, provides methods and sets security to the objects and their associations, with the differential to achieve through additional ability to act in the semantic aspect of the data, not just syntactic.

IV. EXTENSIBLE BUSINESS REPORTING LANGUAGE

The XBRL language is an open and free standard developed by approximately 650 organizations and government agencies. Based on XML, it was designed for the creation, exchange and analysis of financial demonstration in the Internet. As such, it enables investors and professionals of financial market to analyze and extract information in their applications, simplifying one of the key stages of the financial analysis, which consists in the data reentry [21].

XBRL defines the syntax used to report the value of a financial fact based on a set of accounting concepts well-defined within a particular context. XBRL divides the information of financial reporting into two components: instance and taxonomy. The instance is an XML document that contains the facts reported, while the taxonomy is an XML schema that defines the concepts communicated by the facts, beside five *linkbases* which define relationships between elements specified in the schema and between elements in the instance. The combination of an instance XBRL with the schema of its taxonomy and the set of associated *linkbases* constitute an XBRL financial reporting. The following sections describe each of these items in more details.

A. Taxonomy

A XBRL taxonomy corresponds to a XML *Schema* and a set

```
<schema
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xbrli="http://www.xbrl.org/2003/instance">
   <element id="br_assets" name="assets"
xbrli:periodType="duration"
      type="xbrli:monetaryItemType"
substitutionGroup="xbrli:item" nillable="true"/>
   <element id="br_policyCompensation" name="
policyCompensation"
      xbrli:periodType="duration"
type="xbrli:stringItemType"
      substitutionGroup="xbrli:item" nillable="true"/>
</schema>
```

of *linkbases* directly located or referenced in the XML *Schema* [Fallside and Walmsley 2004]. The taxonomy defines the related concepts to a financial report. The XBRL terminology describes a concept as a definition of a reported fact. For example, the asset of a company is denominated by XBRL of concept. The concepts are represented by means of a definition of an element of a XML *Schema* that receives a name and a type. The example displayed in Chart 1 shows an XML Schema from a taxonomy, defining two elements, *br_assets* and *br_ policyCompensation*, and specifying their types.

Chart 1. Example of a document XBRL

Other important component of taxonomy is the *linkbases*. The *linkbases* document the meaning of the concepts, expressing the relationships between them and relating concepts with their documentation. For example, in the *linkbase*, it is defined that the concept is *current assets* is a subsection of the *assets* concept. Relations between XML fragments occur in various forms in XBRL such as relations between an instance and its taxonomy and relationships between financial facts and footnotes that complement the description of these facts. The semantic of the concepts is expressed by a set of relationships which constitute the *linkbases*. XBRL expresses all these relationships using the syntax of simple *links* and extended *links* XLink specification, defined in [DeRose, Orchard and Maler, 2001]. There are five types of extended *links* used in a taxonomy: *definition, calculation, presentation, label* and *reference*. The first three types express relationships between concepts, and the last two express relationships between concepts and their documentations.

Links definitions provide four types of relationships between taxonomy concepts: (1) "general-special" connects a

```
<definitionArc xlink:type="arc"
xlink:from="Postalcode" xlink:to="ZIP"
xlink:arcrole="http://www.xbrl.org/2003/arcrole/genera
l-special" order="1"/>
```

generalist concept to a specialist concept; (2) "essence - alias" establishes an arc between an essential concept and your nickname; (3) "similar - tuples", tuples that have similar definitions are related, even when they have different models of XML content; (4) "requires - element" indicates that the occurrence of a concept implies the obligatory presence of another one. The example illustrated in Chart 2 shows an arc that establishes a generalization-specialization relationship

```
<calculationArc xlink:type="arc"
xlink:arcrole="http://www.xbrl.org/2003/arcrole/sum-
mation-item"
   xlink:from="assets" xlink:to=" assetsCurrency "
weight="1.0" order="1"/>
```

between the concepts "PostalCode" and "ZIP" (PostalCode is a generalization of ZIP). The attribute "order" sets the order of presentation of this *link* for the user.

Chart 2. Example of Link Definition

The *calculation link* is used to establish a sum relationship between concepts, in other words, specifies how the value of a

```
<presentationArc xlink:type="arc"
xlink:from="assets" xlink:to="assetsCurrency"
xlink:arcrole="http://www.xbrl.org/2003/arcrole/pare
nt-child" order="1"/>
```

concept contributes to set the value of another. Chart 3 shows an arc of a *link calculation* in which the concept "assetsCurrency" contributes to the value of "assets" with weight 1.0.

Chart 3. Example of Link Calculation

The *link presentation* defines the hierarchy and order of presentation of the concepts in taxonomy. Chart 4 shows an arc of a *link presentation* that establishes a parent-child type

```
<label xlink:type="resource" xlink:role=
http://www.xbrl.org/2003/role/label
xlink:label="assetsCurrency" xml:lang="en">Assets
Currency</label>
```

relationship in which the concept "assetsCurrency" is the first child of the concept "assets".

Chart 4. Example of Link Presentation

The *link label* expresses relationship between concepts, textual documents and labels. The *link labels* are used to

```
<referenceLink xlink:type="extended"
xlink:role="http://www.xbrl.org/2003/role/link">
<loc xlink:type="locator"
xlink:href="samp001.xsd#s_customerSales"
xlink:label="s_customerSales"/>
<referenceArc xlink:type="arc"
xlink:from="s_customerSales "
xlink:to="s_salesByCustomer_REF"
xlink:arcrole="http://www.xbrl.org/2003/arcrole/conc
ept-reference"/>
<reference xlink:type="resource"
xlink:label="s_salesByCustomer_REF"
xlink:role="http://www.xbrl.org/2003/role/definitionR
ef">
<ref:name>Handbook of Business
Reporting</ref:name>
<ref:pages>5</ref:pages>
</reference>
</referenceLink>
```

provide an explanatory documentation and a set of legible labels to the reader. Chart 5 shows an example of a *link label* in which the concept "assetsCurrency" receives a label "Current Assets". The *xml: lang* attribute specifies the language of the label.

Chart 5. Example of Link Label

Finally, the *link reference* is used to establish a relationship between concepts and normative references in business publications, financial and accounting literature. Chart 6, two references are added to the concept "s_customerSales."

Chart 6. Example of Link Label

B. Instance

The concepts defined in the taxonomy do not contain the values of the financial facts. The values of the facts are informed in the document of XBRL instance. The terminology of XBRL describes these financial facts simply as "facts". The structure of document of a XBRL instance is defined in XBRL *Instance* Document *Schema* that not only defines their attributes and data types, but specifies the basic elements to its composition too. An XBRL instance document may use more than one taxonomy, and the taxonomies may also be interconnected likewise, by extending and modifying each other in various ways. Generally, it is necessary to consider multiple taxonomies related when interpreting an XBRL instance. The set of taxonomies related to a given instance is called a *Discoverable Taxonomy Set* (DTS). The Chart 7 shows an instance document that contains the values for the financial facts: assets, current assets, liabilities and current liabilities.

V. XBRL SUB-REPORT FORMALIZATION

This section describes the concept of XBRL sub-report, which is the basis for the formalization of language XBACL. Before defining the sub-reports XBRL, it is important to present the concept of sub-documents. In the relational model, relational algebra operations [Elmasri and Navathe, 2004] are used to remove columns and rows in order to generate sub-tables. In XML, the generation of a sub-document occurs with the removal of sub-trees [Sahuguet and Alexe, 2005].

Definition 1: Given an "Y" XML document, defined by Y

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
xmlns:xlink="http://www.xbrl.org/2001/XLink"
mlns:link="http://www.xbrl.org/2003/linkbase"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instа
nce"
xmlns:br="http://www.example.com.br">
   <br: assets precision="3" unitRef="u1"
contextRef="c1">6784</br:assets>
   <br:assetsCurrency precision="3" unitRef="u1"
     contextRef="c1">5684</br: assetsCurrency>
   <br:liabilities precision="3" unitRef="u1"
contextRef="c1">635</br:liabilities>
   <br:liabilitiesCurrent precision="3" unitRef="u1"
contextRef="c1"> 235
   </br: liabilityCurrency>
   <context id="c1"><!-- ... --></context>
   <unit id="u1"><!-- ... --></unit>
</xbrl>
```

= (N, H, λ, <, F) where N is a set of *nodes* of document, H is the relationship parent-child of the nodes, λ is the function that associates each node to a label, < is the order relation between document nodes, and F is the subset of N nodes that has no child nodes. A Y' document defined by Y' = (N ', H', λ '<', F ') is considered a Y sub-document if and only if it could satisfy the following assumptions:

(1) N' ⊆ N;
(2) H' '⊆ H;
(3) λ ' = λ;
(4) <' = <;
(5) F' ⊆ F.

Chart 7. Instance Document Example

A q(D) sub-document is the result of the command operation XSQuirrel q( ) on the "D" document [Sahuguet and Alexe, 2005]. The q(D) sub-document is constructed by removing some D nodes, resulting in a D sub-document. This result is used in the following definition.

Definition 2: Given a "R" XBRL report defined by R = (I, T, L, R, D, C), where I is the instance document, T is the XML *Schema* document that defines the taxonomy of the report, L is the label *linkbase*, R is the reference *linkbase*, D is the definition *linkbase* and *C*, calculation *linkbase*. Then a R' report defined by R' = (R ', T', L ', R', D ', C') is a R sub-report if and only if it could satisfy the following assumptions: (1) I' = q(I), in other words, I' is a I sub-document; (2) T' = T; (3) L' = L; (4) R' = R; (5) D' = D; (6) C' = C.

Then, a R' sub-report obtained from a R report is constituted of the same set of elements of taxonomy, and its instance document is a sub-document of the R instance. This definition is the basis for the construction of XBACL language presented in the following section.

VI. XBACL Language

XBACL is a language for specifying access control policies to financial data in XBRL. So, we used the notion of sub-reports to specify the parts of a financial report that a user has access.

An access control policy described in XBACL is specified by P = (U, A, C, T, R), where U is a user or user profile; A is an action, which can be read, updated, deleted and created; C is the set of concept, on which the policy is applied; T is the type of policy that is classified as permission or denial; and R defines if the policy is recursive, in other words, it can also be applied to the concepts of C network relationships. Besides this definition, other important features in building a language for specifying access control policies are described in the following sections.

C. Granularity and Selection Language

In most of the correlated work, the smallest unit of protection is the XML node and XPath is used to identify the nodes on which access control occurs. In XBACL the smallest unit is the sub-document.

The great advantage of using sub-documents as smallest unit of protection is because this makes access control policies concise. Furthermore, the use of XPath does not guarantee that the parts of a document that some user has access form a valid XBRL document. Consequently, this can result in a XBRL document that will not be validated by XBRL processors.

D. Completeness

One of the main features that guarantee a completeness of an access control policy is to ensure that every request has a valid response. Consequently, the first action to do is setting a default value, so that when a concept has not owned policy, it is defined a criterion of conflict resolution. So, XBACL believes that if there is not a rule of access control to a particular node, then the node is considered inaccessible. We chose to apply this policy to access control, because it contributes to the characteristic of least privilege by imposing more rigorous to data access.

XBACL allows that a concept has a policy of denial and access permission to a same user. In this case, denial prevails, in other words, a policy of denial takes precedence over a policy of permission. This option was chosen for safety reasons, because the denial application will not be compromised by the permission application with the propagation applied to a superior item in the hierarchy. However, it is necessary to ensure that every query generate a valid response in XBACL.

E. Recursion

The ability to create recursive access control policies is a feature that allows the construction of more succinct access rules. Unlike the hierarchical relationship, typical of an XML document, a XBRL taxonomy has a set of extended *links* that define a complex network of relationships. The Figure 1 shows some of the possible formation of social networks in a XBRL taxonomy. In a XBRL taxonomy, the arcs of *links definition*, *calculation* and *presentation* organize the financial concepts in relationship networks. For the scope of XBACL, only *links definition*, *calculation* and *presentation* are considered because only these *links* establish relationship between the elements of a XBRL document.

To specify the propagation of policies, it is necessary to define its propagation algorithm. The propagation algorithm All the concepts found in the networks are added to the permission list. In the second stage, the process is similar to that in the first one.of a XBACL policy consists of two main phases. All the concepts that have a policy of denial and who are in the permission list are removed. Then, for each removed concept, the algorithm navigates by the arcs that originate from it, removing from the list, all the navigated concepts. This order is the result of the strategy that the denial prevails, described before. The chart 8 illustrates this propagation algorithm.

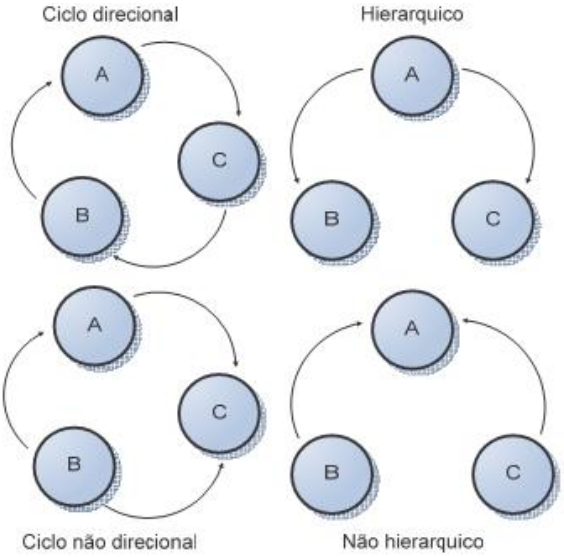

Figure 1. R**elationship networks in a XBRL taxonomy**

```
if (rule.Recursive) {
IList concepts = GetConceptsNet (rule.Concepts);
RemoveConcepts (concepts);
} else { Else {}
RemoveConcepts (rule. Concept);
} } } } } } } }
```

Chart 8. Propagation algorithm

F. Collections of documents

One of the requirements of a policy of access control is the possibility to apply control access policies to specific documents. However, restrict XBACL, permitting your application only to specific documents, makes the policy implantation a complex and laborious process, because for large amounts of documents would be necessary to adopt individualized policy. So, XBACL uses an approach that includes both access control in the level of instances and the access control based on a set of them. In the case of the specification of an access rule to an instance, the solution is simple and only requires the specification of the document to which the policy will apply. In the case of the specification of an access rule in a set of instances, its implementation is more complex and is based on the relationship contained between an instance and it taxonomy. In this case, it will be considered

Value of Attribute	Description of the semantics
http://xbrl/2012/role/ positive_local	Policy that ensures access only to the informed concept.
http://xbrl/2012/role / positive_recursive	Policy that ensures access to the selected concept and all its children.
http://xbrl/2012/role / negative_local	Policy that denies access only to the informed concept.
http://xbrl/2012/role / negative_recursive	Policy that denies access to the selected concept and all its children.

```
<linkbase xmlns="
http://www.xbrl.org/xbrl/2012/xbacl"
xmlns:samp="http://www.xbrl.org/xbrl/sample"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.xbrl.org/xbrl/sample police.xsd"
xml:base="http://www.xbrl.org/xbrl/sample">
<policy xlink:type="resource"
xlink:role="http://www.xbrl.org/xbrl/2012/role/positive_local"
xlink:policy="assets"
xlink:credencial="CIO"></policy>
<policy xlink:type="resource"
xlink:role="http://www.xbrl.org/xbrl/2006/role/positive_recursive"
xlink:policy="liabilities"
```

that the rule applies to a set of structured instances according to a given taxonomy.

G. XBACL Syntax

The extended *links* in taxonomy provide additional information on the existing concepts in it, expressing relationships between concepts or linking them with their documentations. With the goal of maintaining the XBRL standard, XBACL was defined by an extended link, as it can be seen in Chart 9. The *policy* element displayed in this picture is a XBRL extended *link*. Its goal is to relate a concept with a user profile and establish the values needed to define an access policy.

Chart 9. Definition of Policy Element

Each one of the *policy* elements of an access policy must be inserted into a *container* XLink. The XLink *container* must be a *linkbase* element that can be found in the path "schema/annotation/appinfo/*" provided in the taxonomy *schema* or must be the root element from another document. The chart 10 illustrates how the policy element may be used. All the present attributes in the policy element are required.

The role attribute describes the type of access policy. The attribute named policy indicates the concept of XBRL taxonomy to which the policy is applied. The credential attribute identifies the user group, while the type attribute determines if the policy is to allow or deny access, and finally, the last recursive attribute indicates if the policy is recursive, in other words, if it can be applied to all the concepts below the specified concept hierarchy.

```
<schema
targetNamespace="http://www.xbrl.org/xbrl/2012/xbacl"
<Schema targetNamespace =
http://www.xbrl.org/xbrl/2012/xbacl xmlns =
"http://www.w3.org/2001/XMLSchema"
xmlns:link="http://www.xbrl.org/2003/linkbase"
xmlns:xl="http://www.xbrl.org/2003/XLink"
xmlns:xlink="http://www.w3.org/1999/xlink"
elementFormDefault="qualified"> elementFormDefault =
"qualified">
<element name="policy"
substitutionGroup="xl:resource">
<annotation>
<documentation>
Defining of policy element
</documentation>
</annotation>
<complexType mixed="true">
<complexContent mixed="true">
<extension base="xl:resourceType">
<sequence>
<any namespace="http://www.w3.org/1999/xhtml"
processContents="skip" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
<anyAttribute
namespace="http://www.w3.org/XML/1998/namespace"
processContents="lax" />
</extension>
</complexContent>
</complexType>
</element>
</schema>
```

The Table 1 lists all the possible values for the role attribute of the element policy and describe their meanings. Only four items presented in this table are considered valid values for the *role* attribute.

Table 1. Valid Values fo the atributo *role*

Chart 10. Linkbase example with two Policy elements

VII. CASE STUDY

The XBACL language has been implemented in the access control module of XBRL Framework, to specify the system access control policies. This section presents the features of the module of the system access control and shows an example of XBACL language utilization in the creating of new policies. The Figure 3 shows the initial screen of the XBRL access control system administrator.

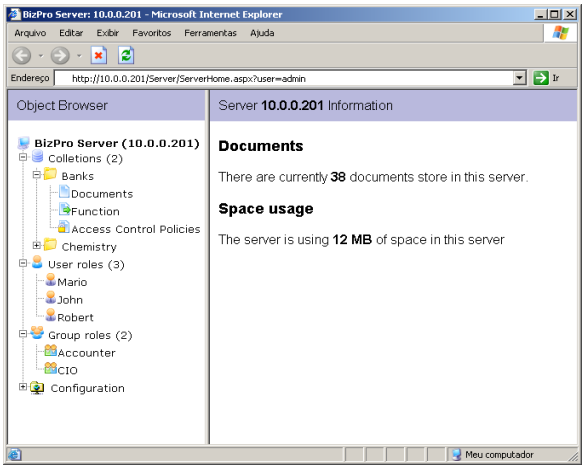

Figure 3. Initial screen management BizPro

This server is instantiated with two document collections: Banks and Chemistry. To access the access control policies of the Banks collection is necessary to click on the link.

After the creation of this new policy, the services (WebService) of the XBRL Framework access control system now apply the new access control specifications in the requested consultations by the participant users of the group *Accounter* and CIO.

On the screen of the system (Figure 4), the access control policies are showed applied to Banks collection, in it, only a policy is applied to the user Mario. Next, the Figure 5 displays the creating screen for a new policy, exemplifying that one contained in Chart 10.

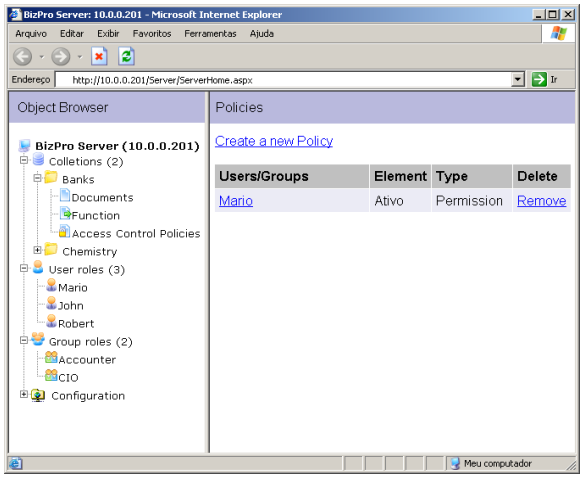

Figure 4. Access control policies list applied to Banks collection

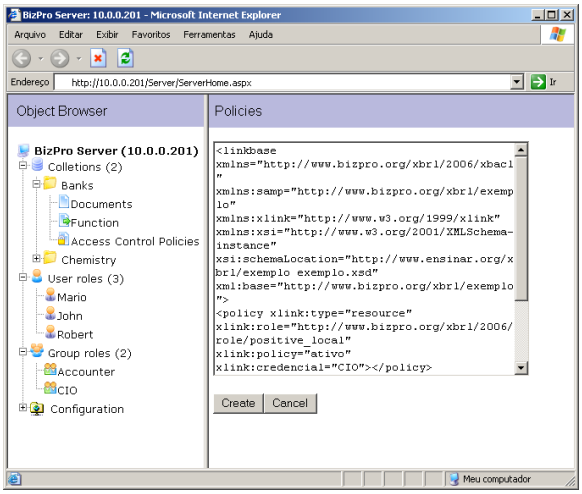

Figure 5. Creating a new access control policy

CONCLUSION

This paper aimed the definition of an access control policy to financial data described in XML language. The main contributions of this work are: (1) Formalizing the concept of XBRL sub-report based on the definition of a sub-document; (2) Definition of a model for access control policies to financial reports; and (3) Specification of the semantics and syntax of the XBACL language.

The XBACL language was validated by implementing the access control module of XBRL Framework. This implementation proved itself satisfactory, enabling the creation of concise access control policies and generating XBRL documents as a valid result.

It is identified the possibility of assessing the application of XBACL for any XML document in a future work. Moreover, there is also the need to investigate the implementation of policies for documents that have different taxonomies.

References

[1] Akker, TVD, Snell, QO and Clement, MJ The YGuard access control model: set-based access control. In *Proceedings of the sixth ACM symposium on Access control models and technologies* (SACMAT '01). ACM, New York, NY, USA, 75-84, 2001.

[2] Bertino , E., Castano, S. and Ferrari , E. On Specifying Security Policies for Web Documents with an XML-based Language, In: *Proceedings of the sixth ACM symposium on Access control models and technologies*, Chantilly, Virginia, USA, p. 57–65, 2001.

[3] Bonatti, P., Vimercati, SC and Samarati, P. A Modular Approach to Composing Access Control Policies. In: *Proceedings of the 7th ACM conference on Computer and communications security*, Athe ns, Greece, p. 164–173, 2000.

[4] Bonatti, P., Vimercati, S. and Samarati, S. A Algebra for Composing Access Control Policies. In: *ACM Transactions on Information and System Security*. ACM, New York, NY, USA p. 1–35, 2002.

[5] Boritz, JE and NO, WG. Security in XML-based financial reporting services on the Internet, In: *Journal of Accounting and Public Policy*, Volume 24, Issue 1, January–February, p. 11-35, 2005.

[6] Damiani, E., Vimercati, S., Paraboshi, S. and Samarati , P. Design and Implementation of an Access Control Processor for XML Documents, In: *WWW9 / Computer Networks, North- Holland Publishing Co*, Amsterdam, The Netherlands, p. 59 – 75, 2000.

[7] Damiani , E. , Vimercati, S., Paraboschi, S. and Samarati, P. A Fine-Grained Access Control System for XML Documents. In: *ACM Transactions on Information and System Security*. ACM, New York, NY, USA Volume 5 Issue 2, May, p. 169–202, 2002.

[8] Derose, S., Maler, E. and Orchard, D. XML Linking Language (XLink) Version 1.1. W3C Recommendation, 2010.

[9] Elmasri, R. and Navathe, SB *Fundamentals of Database Systems*, Pearson Education, 2004.

[10] Engel, P. et al. *Extensible Business Reporting Language (XBRL) 2.1*. XBRL Recommendation, 2003.

[11] Fallside, DC and Walmsley, P. *XML Schema Specification*. W3C Recommendation, 2004.

[12] Fundulaki, I. and Marx, M. Specifying Access Control Policies for XML Documents with XPath. *9th ACM Symposium on Access Control Models and Technologies*, Yorktown Heights, New York, USA, p. 61–69, 2004.

[13] Merkle, Ralph C. One Way Hash Functions and DES. In Advances in Cryptology – CRYPTO'89, *Lecture Notes in Computer Science*. Volume 435, 1990, pp 428-446.

[14] Gottlob, G., Koch, C. and Pichler, R. The Complexity of XPath Query Evaluation. *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, San Diego, California, p. 179–190, 2003.

[15] Gowadia, V. e Farkas, C. RDF Metadata for XML Access Control. *In: Proceedings of the 2003 ACM workshop on XML security*, New York, USA p. 39–48, 2003.

[16] Lim, C.; Park, S. and Son, SH. Access Control of XML Documents Considering Update Operations. *In: Proceedings of the 2003 ACM workshop on XML security*, New York, USA p. 39–59, 2003.

[17] Miklau, G. and Suciu, D. (2003) Controlling access to published data using cryptography. *Proceedings of 29th International Conference on Very Large Database*. p. 898-909, 2003.

[18] Murata, M., Tozawa, A. and Kudo, M. XML Access Control Using Static Analysis. *Proceedings of the 10th ACM conference on Computer and communications security*, Washington DC, USA, p. 73–84, 2006.

[19] Parmar, V., Shi, H. and Chen, S. XML Access Control for Semantically Related XML Documents. *36th Annual Hawaii International Conference on System Sciences*, p. 10-22, 2003.

[20] Sahuguet, A. and Alexe, B. Sub-Document Queries Over XML with XSQuirrel. *Proceedings of the 14th international conference on World*, Chiba, Japan, p. 268–277, 2005.

[21] Silva, P C. *Exploring Markup Languages for the Representation of Financial Reports*, Dissertation (Masters of Science of Computer) – Salvador University, Salvador, 2002.

[22] Takase, T.; Uramoto, N.; Baba, K., "XML digital signature system independent of existing applications," *Applications and the Internet (SAINT) Workshops, 2002. Proceedings. 2002 Symposium on* , vol., no., pp.150,157, 2002.

[23] Bartel, Mark et. AL. W3C Signature Recommendation. *XML Signature Syntax and Processing (Second Edition)*, 2013.

[24] Imamura, Takeshi et. Al. W3C Encryption Recommendation. *XML Encryption Syntax and Processing*, 2013.

[25] Hallam-Baker, Phillip. Shivaram H. Mysore. W3C XKMS 2.0 Specification. *XML Key Management Specification (XKMS 2.0)*, 2005.

[26] R. Bhatti, J. B. D. Joshi, E. Bertino, A. Ghafoor, "Access Control in Dynamic XML-based Web-Services with X-RBAC", In: *The First International Conference on Web Services*, Las Vegas, June 23-26, 2003.

[27] Marshall, B.; Mortenson, K.; Bourne, A.; Price, K. Visualizing Basic Accounting Flows: Does XBRL + Model + Animation = Understanding? *The International Journal of Digital Accounting Research* Vol. 10, pp. 27-54, 2010.

[28] Rosset, Valerio, Filippin, Cleber V., and Westphall, Carla M. Distribution rights for DRM Systems using Standard Security SAML. In: *Proceedings of the ACM workshop on XML security*, Pages 71 – 79, 2002.

[29] Yue-Shen, Gu, Meng-tao, Yong, Gan. Web Services Security Based on XML Signature and XML Encryption. *Journal of Networks*, vol. 5, no. 9, September, 2010.

[30] Ammari, F.T.; Lu, J., Advanced XML Security: Framework for Building Secure XML Management System (SXMS), *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on* , vol., no., p.120,125, 12-14 April, 2010.

[31] Tibor Jager and Juraj Somorovsky. How to break XML encryption. In *Proceedings of the 18th ACM conference on Computer and communications security* (CCS '11). ACM, New York, NY, USA, p. 413-422, 2011.

[32] GESA – *Software Engineering Group*, Salvador University XBRL Framework. http://www.nuperc.unifacs.br/index.php/grupos-de-pesquisa/gesa, 2012.

Authors Biographies

Ivanildo J. S. Aquino Júnior. Holds a graduation degree in Computer Science from Federal University of Pernambuco (2005) and Master's degree in Computer Science from Federal University of Pernambuco (2008). He is currently a Software Engineer Teacher. He has experience in Computer Science with emphasis in Computer Systems.

Cleberton Carvalho Soares. Student of Master's degree in Systems and Computing at Salvador University - UNIFACS. Professor and Coordinator of Technology in Computer Networking Course in Estácio College of Sergipe.

Paulo Caetano da Silva. Ph.D. in Computer Science by Federal University of Pernambuco, 2010; Master's degree in Systems and Computing, by Salvador University, 2003; Graduation degree in Chemical Engineering, by Federal University of Bahia, 1985. He is currently a professor at Salvador University in the Master Program in Systems and Computing and analyst of the Central Bank of Brazil. He has experience in Computer Science with emphasis in Software Engineering, Database and XML, acting on the following topics: OLAP / XML, XBRL, markup language, information systems, web and financial information.

Sidney Viana. Graduation degree in Electrical Engineering by Catholic University of Rio Grande do Sul (1991) and Master's degree in Electrical Engineering by University of São Paulo - USP (1996) (Medical Informatics). Doctor's degree by USP (Computer Engineering). He serves as a visiting professor of postgraduate course at the Polytechnic School of USP (POLI-USP). He currently works as a professor at Salvador University (UNIFACS). Areas of Interest: Database (Relational Database, Object-Relational Database for WEB, Data Modeling and Data Quality) and Software Engineering (Engineering Requirements, Models of Software Development, Software Quality and Project Management).

Valéria Cesário Times. Graduation degree by Catholic University of Pernambuco (1991), Master's degree in Computer Science by Federal University of Pernambuco (1994) and Ph.D. in Computer Science (PhD on Computer Science) - School of Computer Studies at the University of Leeds (1999). He is currently Adjunct Professor III, Federal University of Pernambuco. He has experience in Computer Science with emphasis in Information Systems, acting on the following topics: data warehouse, geographic database, geographic information services, geographic information systems and tools OLAP.