An Attribute-Oriented Model for Identity Management

ARTICLE · JANUARY 2010

DOI: 10.1109/IC4E.2010.55

4 AUTHORS:

Z.P. Jin
Deggendorf University of Applied Sciences
3 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Jian Xu
Hangzhou Dianzi University
18 PUBLICATIONS **34** CITATIONS
SEE PROFILE

Ming Xu
Hangzhou Dianzi University
42 PUBLICATIONS **89** CITATIONS
SEE PROFILE

Ning Zheng
Hangzhou Dianzi University
40 PUBLICATIONS **81** CITATIONS
SEE PROFILE

An Attribute-Oriented Model for Identity Management

Z.P. Jin, Jian Xu, Ming Xu, Ning Zheng
Computer Science Department of
Hangzhou Dianzi Technological University
HangZhou, China, 310085
jin_z_p@hotmail.com, {jian.xu, mxu, nzheng}@hdu.edu.cn

***Abstract*—Secure, interoperable and flexible identity management (IdM) architecture should be designed in large organizations to alleviate "Identity Confusion". Existing network-centric, service-centric and user-centric IdM models are not always efficient during processing and communicating. In this paper, we propose an attribute-oriented IdM model, which adaptively and efficiently considering Requirement Similarity Degree of attributes in identity management. This approach is the first time to balance the requirement from network, service, user and other aspect. We elaborate the design approach of attribute-oriented module and implement it in a digital library case to show its effectiveness.**

Keywords: identity management; attribute-oriented;

I. INTRODUCTION

Nowadays, "Identity Confusion" has become a serious problem in some large organizations. In these loosely coupled organizations, such as enterprises, governments, even universities, the identities [1] of members are distributed in scores of different applications which are decentralized, isolated, non-interactive and executed on heterogeneous platforms. Cost of administration and risk of system security increases dramatically. But the efficiency goes down. One of the ways to improve this phenomenon is to design a secure, interoperable, uniform and flexible identity management (IdM) architecture.

Existing IdM architecture modules can be divided into three distinct types: Network-centric, Service-centric and User-centric [2]. Network-centric perspective is concerned with the hardware and circumstance of network. Service-centric forces on service provider-related aspects and is designed to be cost effective. User-centric has an evident principle: user totally controls his/her identities during the whole life-cycle of identity [1].

No matter which perspective, all of them is biased. They are not balance to the requirement from every aspect. In this paper, we propose an attribute-oriented approach based on attribute's correlation to requirement. Attribute is the fundamental of IdM system. In IdM system, many technologies are based on attribute, such as Attribute-Based Access Control (ABAC) [14], Authentication and Authorization. The issue of IdM indeed is the issue of attributes management. An attribute-centric IdM module can be balance in every aspect of requirement.

To expound the new approach well, the paper is organized as follows: Firstly, in section 2, we discuss the advantages and disadvantages of three distinct existing modules; The following, we illustrate the characteristic of attribute in section 3; Our new approach is explicated in section 4 and a use case is elaborated in section 5; Finally, we give a concise evaluation and conclusion in section 6

II. RELATED WORKS

Network-centric, service-centric and user-centric are three distinct IdM solutions addressing for different requirements. In this section we will introduce them concisely and discuss about the advantages and disadvantages of them.

A. Network-centric Architecture

The rapid development of the network and the ability to deploy software over a network has given rise to network-centric software systems [4]. Network-centric IdM architecture is concerned on network provider-related issues, i.e. network element management, configuration management, network infrastructure security and access control, etc [2]. The advantages are summarized as follows: network-centric IdM architecture reduces the financial cost remarkably with the consideration of the existing infrastructure and maximize reusing of the resources. It also controls the system's interaction of information exchange and ensures the system secure on the transfer layer.

However, the disadvantages are also palpable. Because of the much more attention to the network, the demand of service and user are easily neglected.

B. Service-centric Architecture

A service-centric system focuses on the service provider related aspects and enables to dynamically select and adjust which services to use, such as services security, authentication, authorization, access control and so on [5]. The notion of service-centric is widely recognized. Kerberos [6] is an effective protocol for the authentication service. Project Shibboleth [7] defines standards and specifications for a framework facilitating cross-institutional resource sharing.

Service-centric IdM architecture is designed from the service-provider's (SP) perspective and can be cost effective or easily deployed. The SP controls the management of identities intensively, which makes the

978-0-7695-3948-5/10 $26.00 © 2010 IEEE
DOI 10.1109/IC4E.2010.55

identities maintained securely, easily and effectively. But it ignores the importance of user's participation in the system and is careless about the infrastructure of the whole system which brings inconvenient and poor usability.

C. User-centric Architecture

The user-centric architecture builds on a notion that everything should be under the user's control. It puts user into an important role. The notion is implemented in many new technologies, such as the Security Assertion Markup Language (SAML2.0) [8], the UAC (User Access Control) [9] of Windows Vista and the SUDO [10] of Linux.

User-centric IdM system takes the perspective from user and brings much more convenient. The mechanism that user controls everything helps to get trust from user and upgrade the organization's reputation. It also enhances the privacy protection and raises user's loyalty.

User-centric architecture satisfies user morebut it also brings some problems. Users control the personal information brings a hard work that they should also set the complicated security setting and make the difficult sharing decision. It also brings maintenance problem to the organization because they should get user's consent first.

III. The Characteristic of Attribute

A. Attributes' Inequality of a Person

Attribute is the fundamental of a person in an IdM system. A digitized person is described by several attributes. An attribute can be intrinsic, i.e. that belongs by nature, such as race, eye color, biometrics (e.g. fingerprints), or extrinsic, i.e. acquired from the outside, including family name, first name, address and so on [13]. Some attributes are persistent or long-lived and some are temporary. Examples of persistent attributes include birthday, race and nationality. Examples of temporary attributes can be age, address and weight. Attributes have different purposes and they are not always "equal". Fig 1 shows the inequality of a person's attributes. And the concept of attribute is defined as follows:

Definition 1 *(ATTRIBUTE): An Attribute denoted by a is a <key, value> pair which describes a characteristic of a person or an organization in a specific aspect.*

B. Attributes, Identities and Entities

Attribute, Identity and Entity are three different representations of the characteristic of a person (or an organization) in identity management. Attribute is the fundamental element to describe a person. An Identity is a set of attributes and identifies a person uniquely. An Entity contains many identities and it is an overall profile of a person. Identity and entity are defined as follows:

Definition 2 *(IDENTITY): An* ***Identity*** *is a set of attributes that describes the characteristic properties. An identity can uniquely identify a person or an organization.*

Definition 3 *(ENTITY): An* ***Entity*** *is an overall profile of a person or an organization.*

From the definitions we can see simply that identity is composed by attributes and it also composes the entity. But the relationship of attribute, identity and entity is complicated. An entity may have two different identities. For example, a person can both be a parent or a teacher in a school system. He or she has both a parent identity and a teacher identity. In a system, an identity must only belong to a person. There isn't any identity can represent two different persons because of the identifiability of identity. Fig 2 shows the relationship between entities, identities and attributes.

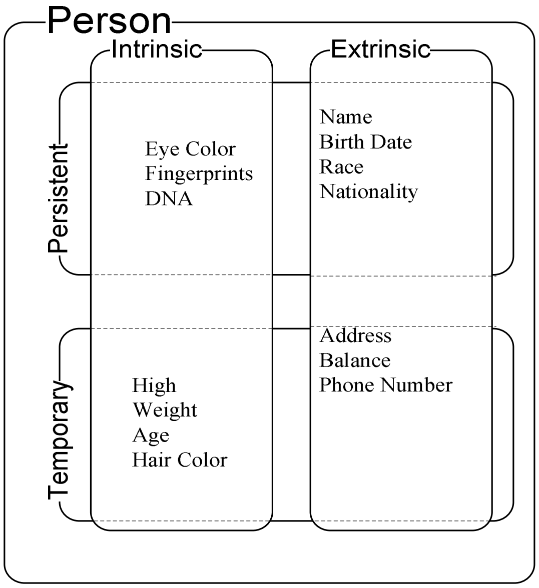

Figure 1. The inequality of a person's attributes

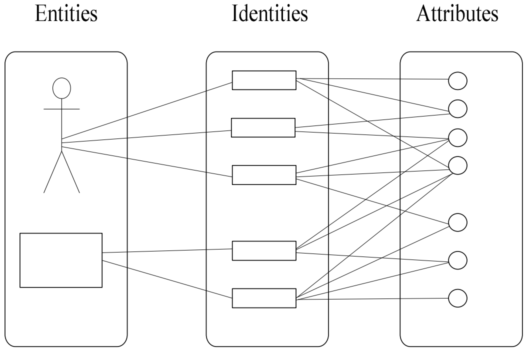

Figure 2. Relationship between entities, identities and attributes

C. Attribute-oriented IdM system.

From the IdM architecture perspective, a system can be subdivided into several small and different functional parts (modules). Module is composed by interoperable components and Component is composed by objects. An object encapsulates a set of attributes. So attributes is the fundamental in IdM architecture and the whole system can be divided into three levels: objects, components, modules. From IdM management perspective, there is also a three levels model. An entity contains several identities and an identity is a set of attributes. Attribute is the fundamental unit.

Fig 3 shows the relationship between IdM architecture and IdM management.

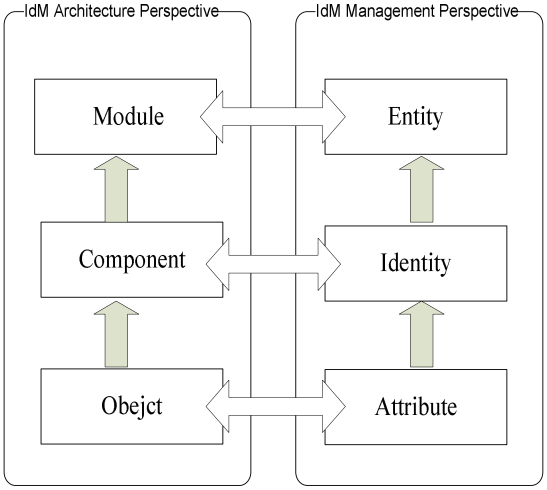

Figure 3. The Relationship between IdM Architecture and IdM Managemen

As we can see, the infrastructure of IdM management are closely related to the IdM architecture.Both of the two models are based on attributes (object is a set of attributes). So an IdM system can be an attribute-oriented system. The issue of identity management is that the issue of attributes management. We can have the definition of attribute-oriented IdM model:

Definition 4 (ATTRIBUTE-ORIENTED IdM MODEL): An ***attribute-oriented IdM model*** *is that an IdM model based on attributes. Both the architecture policy and management policy are based on attributes. Attribute is the fundamental of the whole model.*

IV. AN ATTRIBUTE-ORIENTED IDM ARCHITECTURE APPROACH

In this section, we propose an Attribute-Oriented IdM Architecture approach(AOA) based on the "inequality" of attributes. First, we need to get a attributes list from practical situation.

Definition *5* (ATTRIBUTES LIST): *An* ***attributes list*** *$A(n)$ is a set of all attributes will be used in the system coming from the real requirement of the system. $A(n) = \{a_1, a_2 \cdots, a_n\}$.*

The AOA approach has three phases: (i) The attribute analyzing phase. Analyze the correlation between attributes and requirements. (ii) The component composing phase. Aggregate similar attributes into components. (iii) The system constructing phase. Construct the components into a system using Component-Based Develop (CBD) technology. The processes are elaborated in underlying subsections.

A. The Attribute Analyzing Phase

Attributes have many different characteristic associating with different requirement of IdM. For examples, health record and social security number are more sensitive and have higher privacy requirement than hobby and email address. We capture the difference and it is an aid to decision making when design an IdM system. We need underlying definitions:

Definition 6 *(REQUIREMENT CORRELATION):* ***Requirement Correlation*** *is the correlation between attributes and requirement. Let a_i be an attribute of A(n) (in definition 0) and r_j be the correlation between ai and a certain requirement r_j. The value of correlation can be: low, medium and high which are quantified by 0, 1 and 2. The result is formalized by $R(a_i) = <r_1, r_2, \cdots, r_n>$.*

For instance, there is a system with three requirements: user-interested, security and privacy. Password is high correlative to security and low to privacy. But HealthRecord is medium to security and high to privacy. Both of them are high user-interested. The Requirement Correlation of password is $R(a_{\text{Password}})=<2,0,2>$. and the requirement correlation can be $R(a_{\text{HealthRecord}})=<1,2,2>$. In this phase, we analyze the requirement correlation for every attributes in $A(n)$ and generate an analysis matrix:

$$A(n) = (a_1, a_{2,\cdots,} a_n) \xrightarrow{(r_1, r_2, \cdots, r_m)}$$

$$R(a_1, a_{2,\cdots,} a_n) = \begin{pmatrix} R(a_1) \\ R(a_2) \\ \vdots \\ R(a_n) \end{pmatrix} = \begin{pmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nm} \end{pmatrix}$$

r_{ij} means the degree of the correlation between attribute a_i and requirement r_j.

The analysis matrix $R(a_1, a_{2,\cdots,} a_n)$ denotes the relationship between attributes and requirements.

B. The Component Composing Phase

From the analysis matrix, we know that attributes are associated with requirement. The association is formalized by $R(a_i)=<r_1,r_2,\ldots,r_m>$. But the correlation of some attributes may be similar or even the same. For examples, name, gender and address which are fundamental attributes of a person usually have the same purpose and they will have the similar requirement. Username (or Id) and password are always in pairs. Though password requires higher security and privacy protection, but they have the same purpose for authentication. To judge whether two attributes are similar, we need the following concepts:

Definition 7 *(Requirement Similarity Degree):* ***Requirement Similarity Degree*** *denotes the similar relationship between two attributes quantitatively.*

$$RSD(a_i, a_j) = \frac{1}{m}\sum_{k=1}^{m} r_{ik} \oplus r_{jk} \; (\; r_{ik} \oplus r_{jk} = \begin{cases} 1 & (r_{ik} = r_{jk}) \\ 0 & (r_{ik} \neq r_{jk}) \end{cases} \;)$$

Definition 8 *(Similarity Condition): A Similarity Condition c is a boundary to determine whether the two* *attributes are similar. The condition is a percentage constant of similar attribute in A(n) and decided by the practical requirement of the system. If $RSD(a_i, a_j) \geq c$ then a_i and a_j are similar.*

Definition 9 *(Requirement Similar Set): If $a_1, a_2, \cdots, a_n$ are pairwise similar, then the set S(n)={ $a_1, a_2, \cdots, a_n$ } is a* ***Requirement Similar Set*** *and $a_1, a_2, \cdots, a_n$ are* ***Requirement Similar***.

In the case that HealthRecord and Password, if the similarity condition *c*=0.5 and RSD($a_{\text{HealthRecord}}$, a_{Password})=0.33<0.5, then HealthRecord and Password are not similarity. The Requirement Similar Set Algorithm is used to classify attributes by the requirement similarity.

The result of the algorithm *S*[*j*] is a similar attributes set array. The attributes compose the components.

Requirement Similar Set Algorithm

```
A={ $a_1, a_2, \cdots, a_n$ }; S= O ; j=0;
while (A!= $\varnothing$ )
{
  a=random_pick(A);
  S=S ∪ {a}; A=A-{a};
  for( $a_i$:A )
    if( RSD(a,$a_i$)>c )
      S=S ∪ { $a_i$ }; A=A-{ $a_i$ };
  S[j]=S; j++;
}
return S[j];
```

C. The System Constructing Phase

We construct the components that we get in last subsection into IdM system based on CBD. CBD benefits system to reduce the development cost, time to market and increase flexibility. The crucial of CBD is to have components that are easy to reuse and composition mechanisms that can be applied systematically. [15] The overview of component-based software development process of IdM system is shown in Fig 4.

We create a database to storage the components composed in section 4.2. The CBD processes can be divided into three parts. First, searching and selecting components stakeholder and then integrating and assembling the selected components. Lastly, the new venison should be deployed and evaluated. The detailed process can be seen in [15].

V. A Case Study

To expound the approach, we design a simple case of a digital library of university and set five universal requirements in Table. The digital library has 18 attributes total (after a pretreatment). We also set five requirements: User-trend r_1 means the attributes will be accessed by user frequently. SP-trend r_2 attributes are correlative to service process very much. Security r_3 and Privacy r_4 means the attributes need more in such a requirement. Maintainability r_5 means that the attributes are not static and will cost more financial resource in maintenance. Table Ⅰ shows a part of the attributes list *A*(*n*) in former two rows and analysis matrix $R(a_1, a_2, \cdots, a_n)$ in rest five rows.

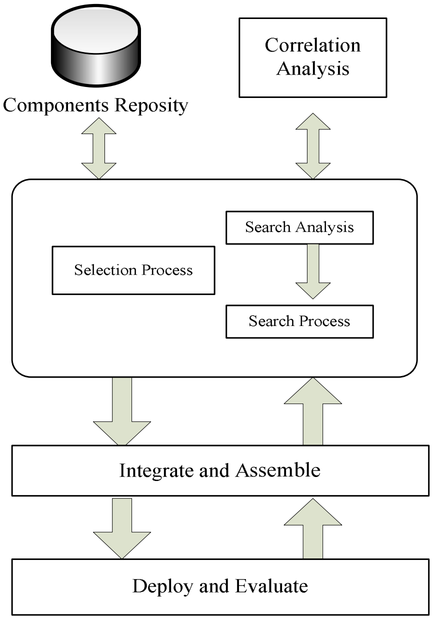

Figure 4. The overview of component-based IdM system develop process

TABLE I. Requirement Correlation Analysis of Digital Library

i	a_i	r_1	r_2	r_3	r_4	r_5
1	RealName	1	1	0	2	1
2	HomeAddr	1	1	0	2	1
3	CampAddr	1	1	0	2	1
4	College	1	1	0	2	1
…	…	…	…	…	…	…
12	ReaderID	0	2	2	0	0
13	Password	2	2	2	0	2
14	Readerlevel	0	2	0	0	0
15	BookedList	2	1	0	1	2
16	BorrowList	2	1	0	1	2
17	BorrowedHistory	2	1	0	1	2
18	IllRecord	2	1	0	2	2

And then in the components composing phase, we classify the A(n) into several RSS (Requirement Similar Set). The similar condition c is preset by 0.5. The result of attributes classification is in Table Ⅱ.

TABLE II. THE RESULT OF AGGREGATE PHASE

Components	RSS	Requirement
Self-service component	a_{RealName} a_{HomeAddr} $a_{\text{CampusAddr}}$ a_{College} $a_{\text{Profession}}$ a_{Gender} $a_{\text{IdentityCardNum}}$ $a_{\text{TelephoneNum}}a_{\text{Email}}$	<1,1,0,2,1>
Authentication component	a_{ReaderID} a_{Password}	<1,2,2,0,1>
Access control component	$a_{\text{CreateDate}}$ $a_{\text{ExpiringDate}}$ $a_{\text{Readerlevel}}$	<0,2,0,0,0>
Library service component	$a_{\text{BookedList}}$ $a_{\text{BorrowedList}}$ $a_{\text{BorrowHistory}}$ $a_{\text{IllLegalRecord}}$	<2,1,0,1,2>

We compose similar attributes and requirement into four components. And then we construct the system through CBD approach. Fig 5 shows the last system we design.

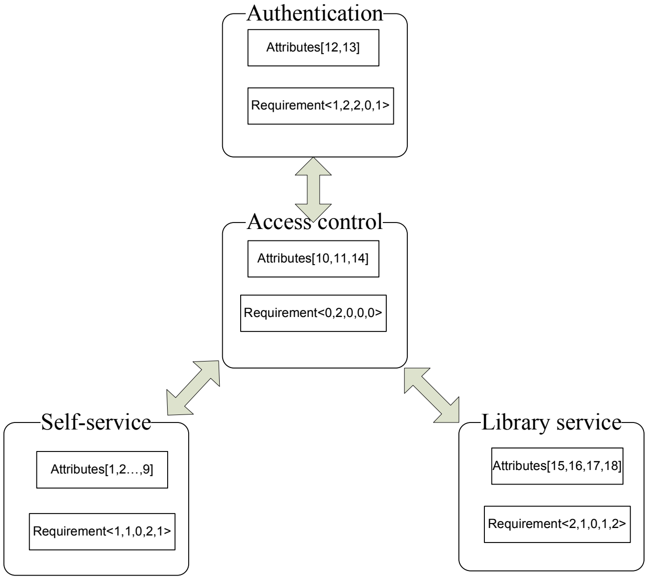

Figure 5. The framework designed by AOA approach

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose an attribute-oriented IdM architecture based on the characteristic of attributes.

Comparing to existing IdM system, attribute-oriented IdM system can be more efficient to the requirement. The requirement is separated by attributes and composed to the components. The result of the approach is an auxiliary decision for designing IdM architecture. And it can help to find a balance between user, SP and other requirement.

However, future study is still needed. Not all of the requirements are correlative with attributes. Some requirements, such as portability and expandability, are not strongly associated to attributes. In the future, more work should be done to this study and to improve the approach.

ACKNOWLEDGMENT

The whole work is supported by Science and Technology Program of Zhejiang Province (NO:2008C21075). Thanks very much.

REFERENCES

[1] P. Windley, “Digital Identity”, O’Reilly, 2005

[2] M.Dąbrowski, P.Pacyna, “Generic and complete three-level Identity Management Model”, The Second International Conference on Emerging Security Information, Systems and Technologies. August 25-31, 2008 Cap Esterel, France

[3] Audun Jøsang, John Fabre, Brian Hay, James Dalziel , Simon Pope1, “Trust Requirements in IdentityManagement”, Proceedings of the 2005 Australasian workshop on Grid computing and e-research – Volume 44, 2005

[4] Amine Chigani, James D. Arthur, and Shawn Bohner, “Architecting Network-Centric Software Systems: A Style-Based Beginning”, The 31st IEEE Software Engineering Workshop. March 6 2007-Feb. 8 2007 Page(s):290 – 299

[5] Laurent Bussard, Elisabetta Di Nitto, Anna Nano, Olivier Nano and Gianluca Ripa, “An Approach to Identity Management for Service Centric Systems”, Springer Berlin / Heidelberg online book, Volume 5377, Page(s):254-265. Dec.11 2008.

[6] Kerberos: The network Authentication Protocol, http://web.mit.edugkerberosgxwhatyis, April 2005.

[7] Project Shibboleth, “Shibboleth Overview and Requirements”, http://shibboleth.internetj.edugdocsgdraft-internetj-shibbolethrequirements-kR.html#abstract, February 2001

[8] S. Cantor et al., Assertions and Protocols for the OASISSecurity Assertion Markup Language (SAML) V2.0, OASIS SSTC, March 2005. Document ID samlcore-2.0-os. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf. 02.2009.

[9] Edward Ray and E.Eugene Schultz, "An early look at Windows Vista security", Computer Fraud & Security,, Vol. 2007, Issue 1, Jan 2007, pp. 4-7.

[10] Mick Bauer, "Paranoid penguin: Linux filesystem security, Part 11", Linux Journal, Vol. 2004, Issue 127, Nov. 2004, pp. 56-67

[11] Tim O'Reilly, "What is Web 2.0", Sep. 2005, http:/ /www. oreilly.com/go/web2

[12] Wikipedia: Component-based software engineering URL: http://en.wikipedia.org/wiki/Component-based_software_engineering

[13] D.V. Thanh, I. Jorstadt, “The Ambiguity of Identity”, Teletronikk, Vol.3, 2007

[14] E Yuan, and J Tong. “Attribute Based Access Control (ABAC) for Web Services”, In proceedings of the IEEE Conference on Web Services (ICWS’05), Orlando Florida, USA. July 2005

[15] Mahmood, S.; Lai, R.; Kim, Y.S. “Survey of component-based software development”, Software, IET, Volume 1, Issue 2, April 2007 Page(s):57 - 66