A Semantic Search Conceptual Model and Application in Security Access Control*

Kunmei Wen, Zhengding Lu, Ruixuan Li, Xiaolin Sun, and Zhigang Wang

Internet and Distributed Computing Lab,
College of Computer Science and Technology,
Huazhong University of Science and Technology,
Wuhan 430074, Hubei, P.R. China
kunmei.wen@gmail.com

Abstract. We propose a conceptual model for semantic search and implement it in security access control. The model provides security access control to extend the search capabilities. The scalable model can integrate other ontology providing the general ontology as the transformation interface. We combine text Information Retrieval (IR) with semantic inference in the model. So it can not only search the resources and the relationships between them according to the user's privileges, but also locate the exact resource using text IR. We build a security ontology based on Role-Based Access Control (RBAC) policy. A semantic search system Onto-SSSE is implemented based on the model. The system can perform some complex queries using ontology reasoning, especially about association queries such as the relationships between resources. The evaluation shows that the new system performs better than exiting methods.

1 Introduction

Semantic Web [1] proposed by Tim Berners-Lee is the next generation of web portals. The aim is to annotate all the resources on the web and establish all kinds of semantic relationships between them understandable for the machine. As the most important application of Semantic Web, semantic search is being got more and more attention. The concept of semantic search is put forward in [2]. Semantic search integrates the technologies of Semantic Web and search engine to improve the search results gained by current search engines and evolves to next generation of search engines built on Semantic Web.

Semantic search finds out the semantic information by means of inferring internal knowledge in Knowledge Base (KB). Description Logic (DL) [3,4] is well known as the base of ontology language such as Web Ontology Language (OWL) [5]. All modern DL system are implemented based on tableaux algorithm [6], many optimized technologies [7] are explored. [8] defines the search object

* This work is partially supported by National Natural Science Foundation of China under Grant 60403027, Natural Science Foundation of Hubei Province under Grant 2005ABA258, Open Foundation of State Key Laboratory of Software Engineering under Grant SKLSE05-07, Huawei Foundation.

R. Mizoguchi, Z. Shi, and F. Giunchiglia (Eds.): ASWC 2006, LNCS 4185, pp. 366–376, 2006.
© Springer-Verlag Berlin Heidelberg 2006

of semantic search. One is searching the Internet. The other is searching the Semantic Web portals. Semantic Web portals are composed of domain ontology and KB. An enhanced model for searching in semantic portals is proposed in [9]. The model combines the formal DL and fuzzy DL [10] to implement the integration of information retrieval and structure query.

Ranking the search results [11,12] is the key technology of semantic search. Since it is expected that the number of relationships between entities in a KB will be much larger than the number of entities themselves, the likelihood that Semantic Association searches would result in an overwhelming number of results for users is increased, therefore elevating the need for appropriate ranking schemes. In [13], a method is proposed to rank the results according to the important values of web resources based on the technology of modern IR [14]. The ranking method in [15] focuses on the semantic metadata to find out the complex relationships and predict the user's requirement to distinguish semantic associations.

Role-based access-control (RBAC) models show clear advantages over traditional discretionary and mandatory access control models with regard to these requirements. There has been a lot of research into the various ways of specifying policy in the last decade. One of them is ontology-based approach. Some initial efforts in the use of Semantic Web representations for basic security applications such as access control of policy have begun to bear fruit. KAoS [16] and Rei [17] are semantic policy languages represented in OWL to specify security policy supporting positive and negative authorization and obligation. The reasoning of KaoS policy is based on DL, which limits the expressive power of policy, as DL doesn't support rule now. As to Rei, it doesn't support the model of RBAC96 explicitly. Besides, they can't intuitively specify the important security principle, separation of duty.

There are great demands for this kind of semantic search considering security issues, such as Intranet search which must satisfy access control request in the back-ground of government or business. We propose a semantic search model that enables the user to find his resources based on his privilege. The proposed model combines text IR with semantic inference. Based on the model a semantic search system Onto-SSSE is implemented and evaluated.

The rest of the paper is organized as follows. We present the architecture of the semantic search model and discuss the components of the model and the relation-ships between components in section 2. The third section discusses the integration of search and inference to get the semantic information and presents the ranking method in semantic search. After that in the forth section the security ontology based on RBAC [18] policy is introduced and instances are described. In section 5 experiment and evaluation are carried out. Related work is introduced in section 6. Section 7 contains conclusions and future work.

2 Architecture of Semantic Search Model

In this section we propose the architecture of the semantic search conceptual model. The architecture of the model is shown in Fig.1. The components of the model and the relationships between them are described as follows.

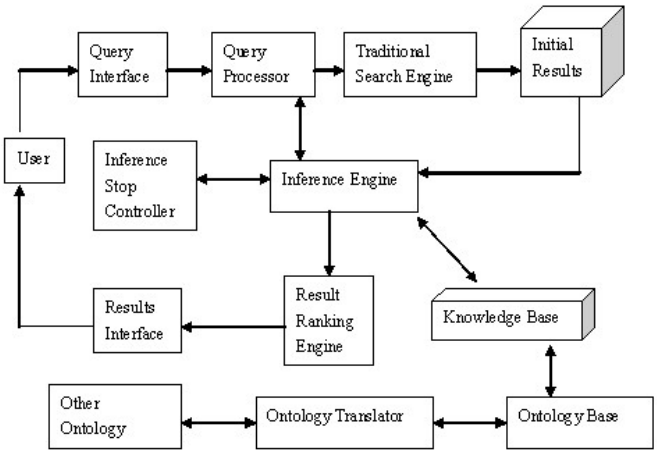

Fig. 1. Architecture of the proposed semantic search conceptual model

Query Interface receives the queries from users. The query is defined as keywords or formal queries. *Query Processor* converts user's queries to uniform format which is defined by the model. Then these queries will be distributed in two ways. One is forwarded to a traditional search engine. The other is forwarded to an inference engine. By means of the operation of *Traditional Search Engine*, we will get the Initial Results using text IR technology. The initial search results are also transformed to inference engine. If the user submits a formal query, then the query will push directly to inference engine. *Knowledge Base* restores domain ontology and reasoning rules or knowledge and is the base for reasoning. *Inference engine* performances the operation of reasoning to get the semantic information and obtains all the search results. *Inference Stop Controller* decides how much to reason and when the reasoning should stop. *Result Ranking Engine* ranks all the results returned by the inference engine. Finally user gets the results through *Results Interface*.

The rest three modules are *Other Ontology*, *Ontology Translator* and *Ontology Base*. They are used to expend the capabilities of semantic search and implement the scalability of the model.

3 Semantic Search Model

The semantic search model mainly is made up of three parts: definition of query form, reasoning based on description logic and result ranking.

3.1 Definition of Query Form

Different users have different privileges for different resources. Some users have the privileges to see or edit or delete the resources such as web pages or news, while others have not the privileges to browse them. Only after assuring that

the user has the right privilege, we could return the resources back to the user through traditional IR technology.

A query is defined as the form $Q_i = Q_{i1} \cap Q_{i2} \cap Q_{i3}$ the semantic search model. Here Q_{i1} means user or role, Q_{i2} is any formal query about resources or the relation-ships between them and Q_{i3} is a keyword query. If Q_{i1} is not appear, that means the user has the default privilege. Q_{i1} and Q_{i2} are implemented based ontology reasoning while Q_{i3} is carried out through traditional text IR technology.

So there are five typical queries as follows:

Q_{i11}: User Query, form as Q_{i11} = "A" where A means a user.

Q_{i12}: Role Query, form as Q_{i12} = "B" where B means a role. In fact, Q_{i11} and Q_{i12} belong to concept query Q_{i1}, so we can get $Q_{i1} = Q_{i11}$ or Q_{i12} = "C" where C means a concept.

Q_{i2}: Relationship Query, form as Q_{i2} = "$C1$"&"$C2$" where C1 and C2 are concepts.

Q_{i3}: Keyword Query, form as Q_{i3} = "D" where D means a keyword which appears in the text. In fact, Q_{i3} belongs to traditional query.

$Q_{i1} \cap Q_{i3}$: Conjunctive Query, form as $Q_{i1} \cap Q_{i3}$ = ("A"or"B")"D" where A means a user, B means a role and D means a keyword.

3.2 Reasoning Based on Description Logic

We implement four kinds of reasoning based on Description Logic in the semantic search model. The architecture of the Knowledge Base based on Description Logic is showed in Fig.2.

The first is *Role Activation Reasoning*. Given $Q_i = Q_{i11}$ where Q_{i11} means user, we can get all the roles the user has. For example if Alice is a user and she can act as Direct or ProjectLeader, then we get all her roles through role activation reasoning.

The second is *Role Privilege Reasoning*. Given $Q_i = Q_{i12}$ where Q_{i12} means role, we can get all the sub-roles of the role and then get all the privileges from these roles. For example if we get role ProjectLeader, through role privilege reasoning we can get the sub-roles including ProductionEngineer and QualityEngineer, so Project-Leader should have all the privileges both ProductionEngineer and QualityEngineer have.

The third is *Relationship Reasoning*. Given $Q_i = Q_{i2}$ where Q_{i2} includes two concepts, we can get the relationship between them or null if there is not any relation-ship. For example if ProjectLeader is the senior role of ProductionEngineer, given the query ProjectLeader & ProductionEngineer, we should be returned the result seniorRoleOf.

The forth is *Conjunctive Query Reasoning*. In fact it integrates inference with search by providing both formal query and keyword query. Given $Q_i = Q_{i1} \cap Q_{i3}$ where Q_{i1} means user or role and Q_{i3} is a keyword query, the semantic search model firstly performance Q_{i1} to judge the user's or the role's privilege. If the user or the role has the corresponding privilege the model carries out Q_{i3} to locate the exact resource. So it can not only locate the exact place of the resource

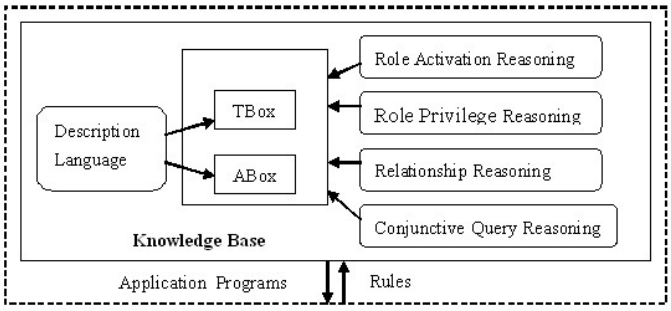

Fig. 2. Architecture of the Knowledge Base based on Description Logic

using the traditional text IR but also implement security access control through inference.

3.3 Result Ranking

Ranking the search results is very important for the implementation of semantic search. It is possible that the number of relationships between entities in a KB will be much larger than the number of entities themselves. We provide a ranking scheme based on the ranking value. The Ranking value for the query Q_i is defined as the form $R_i = R_{i1} + R_{i2} + R_{i3}$ for the query $Q_i = Q_{i1} \cap Q_{i2} \cap Q_{i3}$. Here R_i1 is the ranking value for Q_{i1}, at the same time Ri2 is the value for Q_{i2} and R_{i3} is that for Q_{i3}. The reasoning result is used to compute the values of R_{i1} and R_{i2}. Given Q_{i1}, if the user has the privilege for the resource, then the value of R_{i1} is 1. Otherwise it is 0. If $R_{i1} = 0$ then $R_{i2} = R_{i3} = 0$. That means if the user has no corresponding privilege he will not be permitted to do any operation on the resource, in this case $R_i = 0$.

For R_{i2}, it is possible to return many relationships between two concepts. So the value R_{i2} is determined by the important value of the relationship. For every relation-ship in domain ontology we define an important value Ii which is between o and 1. So it is reasonable to get the conclusion $R_{i2} = I_i$.

R_{i3} is corresponding to Q_{i3}. Searching is used to locate the resource through key-word query. Therefore we can use traditional tf-idf method to compute the value of R_{i3}.

4 RBAC Security Ontology and Description of Instances

KAoS and Rei mentioned above are semantic policy languages represented in OWL to specify security policy. The reasoning of KaoS policy is based on DL. As to Rei, it only supports the rule. KAoS and Rei don't support the recursive authorization. Be-sides, they can't intuitively specify the important security principle, separation of duty (SoD). RBAC is a popular security policy. Here we assume that the readers are familiar with the RBAC policy. We build a security ontology shown in Fig.3 based on RBAC policy.

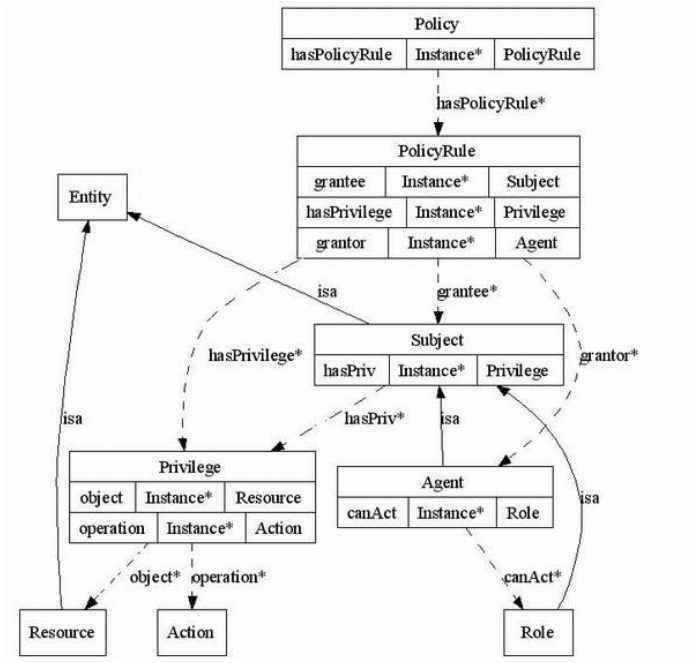

Fig. 3. RBAC security ontology

In RBAC security ontology, nine basic classes are created. They are *Policy*, *PolicyRule*, *Priviledge*, *Entity*, *Resource*, *Agent*, *Subject*, *Role* and *Action*. We give properties for these classes, for example on the top of the figure 3 *hasPolicyRule* is the property of the class *policy*. The right side of the property is its range, for example the range of the property *grantor* is the instances of the class *Agent* and its domain is the class *PolicyRule*. The arrow between two classes indicates the relationships between them. Real line is the subsumption relationship while dashed one defines the property between them. For example *subject* is a subclass of entities, so the relationship between them is "isa". From the figure 3, we can see there are relationships between these classes: *PolicyRule*'s grantee is *Subject*, its grantor is *Agent* and it has *Privilege*; both of *Agent* and *Role* are subclasses of *Subjects*, at the same time *Subject* and *Resource* are subclasses of Entities; *Privilege*'s object is *Resource* and its operation is *Action*. *Agent* can act as *role* where we can think *Agent* has the same meaning with user.

We use OWL DL as our ontology language. As one of W3C's standards, OWL DL is widely used in application. Here is the example fragment of the owl language building the security ontology, showing as the follows:

$< owl : Classrdf : ID = "Subject" >< rdfs : subClassOf >$
$< owl : Classrdf : ID = "Entity"/ >< /rdfs : subClassOf >$
$< /owl : Class >< owl : Classrdf : ID = "Role" >$
$< rdfs : subClassOfrdf : resource = "\#Subject"/ > ..$

$</owl:ObjectProperty></owl:ObjectPropertyrdf:ID="hasPriv">$
$<rdfs:rangerdf:resource="\#Privilege"/>$
$<rdfs:domainrdf:resource="\#Subject"/></owl:ObjectProperty>$

The reason that we choose RBAC policy as our ontology is that the RBAC is more general than other security policies. It is easy to transform other policies to RBAC policy, such as *Mandatory Access Control* and *Discretionary Access Control*, while the reverse transform is not possible. It means that RBAC security ontology is appropriate to be a uniform security policy interface.

To illustrate semantic search more clearly, we give role instances hierarchy graph shown in figure4 and simple privilege instances graph showed in figure 5. From the Fig.4 we can see that Director is the most high-level role.

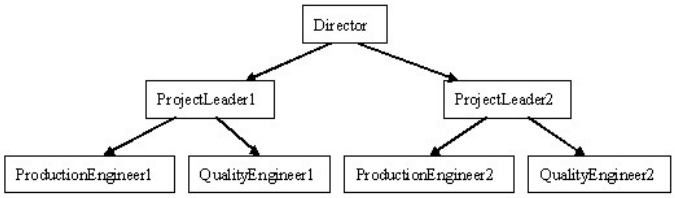

Fig. 4. RBAC security ontology

We create some instances for classes such as roles, agents and resources. There are six roles including *director, ProjectLeader1, ProductionEngineer1, QualityEngineer1, ProjectLeader2, ProductionEngineer2* and *QualityEngineer2*. There are two subclasses of resources resources1 and resources2. We define resource1 two instances webpage11 and webpage12, define resource2 two instances webpage21 and webpage22. We define only one instance "browse" for Action.

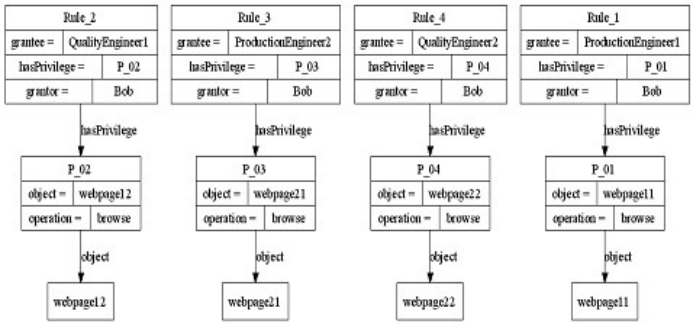

Fig. 5. Simple privilege instances graphy

There are application privileges shown in Fig.5. For example ProductionEngineer1 can browse the resource webpage11 which belong to resource1 and ProductionEngineer2 can browse the resource webpage21 which belong to resource2.

5 Experiment and Evaluation

We implement Ontology Security Semantic Search Engine (Onto-SSSE) in Java. We used the Lucene [19] search engine as the traditional search engine based on key-word query and Jena as the reasoning tool based on RBAC security ontology. We do some experiments on Onto-SSSE. The Table 1 shows the search results for some typical queries.

Table 1. Semantic search results

Query ID	Query form	Query form	Reasoning Type	Query Result
Q_1	Q_{i11}	"Alice"	Role Activation Reasoning	Director, ProjectLeader1
Q_2	Q_{i12}	"Director"	Role Privilege Reasoning	Sub-roles:ProjectLeader1,ProductionEngineer1, QualityEngineer1,ProjectLeader2, ProductionEngineer2,QualityEngineer2; Privileges:(browse,webpage11), (browse,webpage12), ……
Q_3	Q_{i2}	"ProjectLeader1"& "ProductionEngineer1"	Relationship Reasoning	seniorRoleOf
Q_4	Q_{i3}	"computer"	No Reasoning	Null (no privilege)
Q_5	$Q_{i1} \cap Q_{i3}$	"Director & computer"	Conjunctive Query Reasoning	webpage list: webpage11, webpage21…. Where include the text "computer" in these web pages

Q_1 is a simple query just for the user. Q_1 = "Alice". The result is Director and ProjectLeader1, because they are the roles as which Alice can act. Q_2 is a query for the role Q_2 = "Director", we are returned all the sub-roles of Director and all the privilege these roles have. Director has six sub-roles such as ProjectLeader1 and ProductionEngineer1; Director has the privilege (browse, webpage11), (browse, webpage12) and so on. Q_3 is a query for the relationship. The results is seniorRoleOf between ProjectLeader1 and ProductionEngineer1.

Q_4 is the simple keyword query, because the default user or role has no required privilege, so Null is returned. Q_5 is a Conjunctive Query as the form "Director & computer", the result returned is the webpage list where the pages include the text "computer".

As pointed out in [20], currently there is no commonly agreed evaluation method-ology and benchmark for semantic search. We constitute our research group's evaluation dataset. The results are analyzed positively in 90%. The dataset is made up of the RBAC security ontology (including 12 classes, 16 properties and 20 individuals) and the set of campus web pages (more then 200MB). We mainly compare our system with traditional method based on keyword query shown in Table 2. From the table, we can find that the new semantic search system performs better than traditional one especially about the reasoning function.

Table 2. Compare between the semantic search with traditional method

Query form	Reasoning Type	Traditional method	Semantic search
Q_{i11}	Role Activation Reasoning	Not support	Support
Q_{i12}	Role Privilege Reasoning	Not support	Support
Q_{i2}	Relationship Reasoning	Not support	Support
Q_{i3}	No Reasoning	Support	Support
$Q_{i1} \cap Q_{i3}$	Conjunctive Query Reasoning	Not support	Support

6 Related Work

Tap Knowledge Base (KB) [21] is implemented by Stanford University, IBM and other research institutions. Tap KB brings Semantic Web technology into Google to improve the search efficiency through providing additional results. The two kinds of different results are shown on the same page. However the search object is still the traditional resource, not the one on Semantic Web. The method only responds the keyword query, not supporting the form query, so it could not integrate information retrieval and formal semantic query tightly. [22] provides an ontology-based information retrieval model to support result ranking. The method transforms the key-word query to structure query, not combining them.

Swoogle, a prototype system of IR is provided in [23]. The search results are physical documents on Semantic Web (such as RDF and OWL files). However Swoogle has not used the semantic structure information in documents. When the large documents are queried, the useful information is very little and user need analyze the whole file to locate the semantic information.

Turing center in the University of Washington develops the system KnowItAll [24] to extract the information on the Web. [25] prefer some methods of information extraction to search the Web and build up domain KB. Its long-term aim is to re-place the search engine by information extraction. This is another kind of semantic search.

7 Conclusions and Future Work

In this paper we propose a conception model for semantic search and apply it in security access control domain. We combine text IR with semantic reference in the model. The model extends the search capabilities of existing methods through implementing security access control. It also can answer some complex queries such as the relationships between resources. A semantic search system is implemented based on the model. The evaluation shows that the new system performs better than the exiting methods.

We plan to get improvement in the following three aspects. The first is to perform search in a larger dataset. The second is to improve the reasoning efficiency. The reasoning efficiency can not satisfy the user.

References

1. T.Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. Scientific American, May 2001
2. Guha R, McCool R, Miller E. Semantic search. Proceeding of the 12th International World Wide Web Conference. Budapest, Hungary, May 2003: 700-709
3. Franz Baader, Deborah McGuinness, Daniele Nardi, et al. The Description Logic Hand-book: Theory, Implementation and Applications, Cambridge, UK: Cambridge Univ. Press, 2003
4. D. Calvanese, G. Giacomo, and M. Lenzerini. Ontology of Integration and Integration of Ontologies. In Description Logic Workshop 2001: 10-19
5. Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL: The Making of A Web Ontology Language. J. of Web Semantics, 2003, 1(1):7-26
6. Ian Horrocks and Ulrike Sattler. A Tableaux Decision Procedure for SHOIQ. In Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI), 2005
7. F. Baader and U. Sattler. An Overview of Tableau Algorithms for Description Logics. Studia Logica, 2001, 69:5-40
8. A. Sheth, C. Bertram, D. Avant, B. Hammond, K. Kochut, and Y. Warke. Managing Se-mantic Content for the Web. IEEE Internet Computing, 2002, 6(4)
9. Lei Zhang, Yong Yu, Jian Zhou, Chenxi Lin, Yin Yang: An Enhanced Model for Searching in Semantic Portals. WWW 2005: 453-462
10. U. Straccia. Reasoning Within Fuzzy Description Logics. Journal of Artificial Intelligence Research, 2001(14)
11. N. Stojanovic, R. Studer, and L. Stojanovic. An Approach for the Ranking of Query Re-sults in the Semantic Web. In Proc. of ISWC 2003
12. Anyanwu, K., Maduko, A., and Sheth, A.P.: SemRank: Ranking Complex Relationship Search Results on the Semantic Web, Proceedings of the 14th International World Wide Web Conference, ACM Press, 2005
13. Bhuvan Bamba, Sougata Mukherjea: Utilizing Resource Importance for Ranking Semantic Web Query Results. SWDB 2004: 185-198
14. Baeza-Yates and Ribeiro-Neto. Modern Information Retrieval. Addison Wesley 1999
15. Boanerges Aleman-Meza, Christian Halaschek-Wiener, I. Budak Arpinar, Cartic Rama-krishnan, Amit P. Sheth, Ranking Complex Relationships on the Semantic Web, IEEE Internet Computing, 2005, 9(3): 37-44

16. A. Uszok, J. Bradshaw, R. Jeffers, et al. KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement. IEEE 4th International Workshop on Policies for Distributed Systems and Networks, 2003
17. L. Kagal, T. Finin, and A. Joshi. A Policy Language for Pervasive Systems. Fourth IEEE International Workshop on Policies for Distributed Systems and Networks, 2003
18. Ravi S. Sandhu, Edward J. Coyne et al. Role-Based Access Control models. IEEE Com-puter, 1996, 29(2): 38-47
19. Lucene Search Engine. http://jakarta.apache.org/lucene
20. C. Rocha, D. Schwabe, and M. P. de Arag ao. A Hybrid Approach for Searching in the Semantic Web. In Proc of WWW 2004: 374-383
21. Guha, R., McCool, R.: TAP: A Semantic Web Test-bed. Journal of Web Semantics, 2003, 1(1)
22. Vallet D, Fernmndez M , Castells P. An Ontology-based Information Retrieval Model. 2nd European Semantic Web Conference (ESWC). Heraklion, Greece, May 2005
23. Ding L , Finin T, Joshi A, et al. Swoogle: A Search and Metadata Engine for the Semantic Web. In CIKM'04. Washington DC, USA, November 2004
24. Michael Cafarella, Doug Downey, Stephen Soderland, and Oren Etzioni. KnowItAll: Fast, Scalable Information Extraction from the Web. Proceedings of the Conference on Empiri-cal Methods in Natural Language Processing EMNLP 2005
25. Ana-Maria Popescu and Oren Etzioni, Extracting Product Features and Opinions from Reviews, Proceedings of the Conference on Empirical Methods in Natural Language Proc-essing EMNLP 2005