DOI 10.1007/s11416-017-0301-y

ORIGINAL PAPER

Secure access policy for efficient resource in mobile computing environment

Sun-Moon Jo[1]

Received: 10 March 2017 / Accepted: 5 June 2017 / Published online: 30 June 2017

Abstract As the Internet was activated and the mobile environment developed, it has become more common to access dynamic XML data regardless of location and time. XML is widely used for information exchange and representation of data for databases, applications, etc., using the advantage to describe information. As a result, large-capacity XML data becomes increasingly complex, and demand for data access policies is increasing. Security issues such as authorization of access to resources, authentication, security enhancement and privacy arise. The mobile computing environment differs from existing information systems in several ways, so it is difficult to apply the existing access control as it is. Therefore, this paper proposes a secure access policy method for query processing to enable efficient resource management in dynamic XML data environment. The results of the evaluation are also presented to show that the additionally proposed method is efficient and excellent.

Keywords XML · Access control · Policy · Security · Mobile computing

1 Introduction

As the web is activated and the mobile environment develops, it is written and transformed in XML form in various fields using XML, which is a standard of data exchange, so the need for efficient resource management and security of data is recognized as important. Since information is distributed and shared on the network, large capacity XML is not safe from unauthorized access to information and forgery. In this mobile environment, safe information access management policies are needed in terms of security and privacy [1].

In early days, several approaches to information were developed. Access Control Lists (ACL), Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role-Based Access Control (RBAC) are representative [2,3]. The Access Control List is used for Windows or UNIX and can be accessed according to the permissions granted to the user. Discretionary Access Control has an owner for each information object. Owners can autonomously grant or collect the authority to or from other users. The Mandatory Access Control is used in an environment where strict information protection is required. The security level is assigned to the user and the information object, and the user is allowed to access to information appropriate to his/her security level. The RBAC differs from the way in which the authority was directly assigned to existing users. The authorization rights are assigned to roles according to current business performance. Also, it supports users to be able to manage rights by allowing them to belong to the role.

The authority is assigned so that the user can access and use only specific items of XML data for large-capacity XML data just as it gives users access to information from existing databases in the mobile computing environment. To do this, the user's access authority must be manageable. When a user accesses large capacity XML data, it must be able to control according to secure authority. A simple way is to have a separate document according to the authority. This makes it difficult to change the data due to waste of storage space and duplication of documents. Therefore, unauthorized parts are removed from the user's authority, so that the part where secure access is permitted is transmitted as information.

✉ Sun-Moon Jo
sunmoonpink@hanmail.net

[1] Department of Computer Information Technology Education, Pai Chai University, 155-40 Baejae-ro, Seo-Gu, Daejeon, Korea

The XML digital signature supporting the digital signature generation and verification functions for XML data defines the security method. It is an XML-based digital signature technique that can generate and verify digital signature in XML form for various types of electronic documents such as digital contents including XML. It can provide information protection functions such as authentication, integrity and non-repudiation for electronic documents etc [4].

Before developing a secure access policy for mobile computing environments, this paper examines the difference of access control in mobile computing environment and describes requirements for accessing large capacity XML data in mobile computing environment. This makes the specification of access to large capacity XML data important, requiring requirements for secure access systems. First, several levels of secure protection should be supported because the large capacity XML data contains a variety of information. Second, large capacity XML data does not always match with the predefined data type. The policy needs details in relation to the data type, so the situation that is not covered by the existing approach should be managed in detail.

This paper is organized as follows. In Sect. 2, we examine related research and problems of existing approaches. Section 3 describes a resource-efficient secure access policy for large-capacity XML data in a mobile computing environment. In Sect. 4, we describe through comparison with the characteristics of traditional techniques. In addition, we describe the large-capacity XML data performance evaluation with respect to the proposed Resource Efficient Secure Access Policy (RESAP). Finally, Sect. 5 describes the conclusion and future work.

2 Related work

2.1 Related work

Based on the paper [5,6], we describe XML document according to DTD format. A department is an outer element that contains all elements of a document element. The document provides the history, salaries, and medical records of employees. Attributes are in the form name = value. $\mathcal{LE}$ is the set of element identifiers. Label is an element tag and an attribute name. Figure 1 shows a graphical representation of the XML documents. XML document is a graph representing elements, attributes, and edges between them. A node representing an attribute is displayed in a circle with its value. The graph contains an edge that represents the relationship between element attribute and element subelement, and a link edge that represents the link between the elements introduced by the IDREF type attribute.

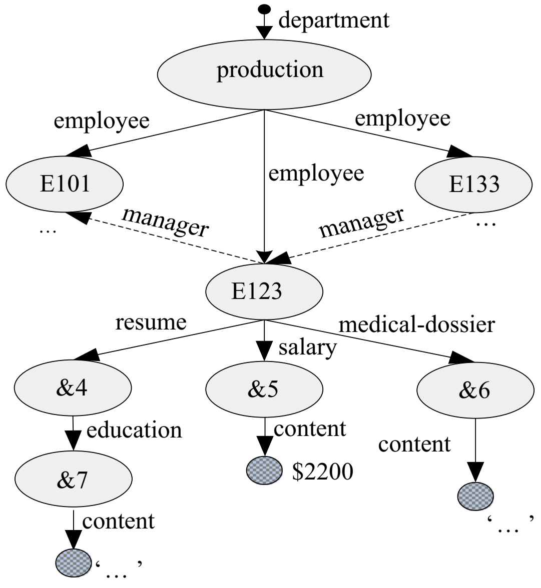


Fig. 1 Example of XML graph representation

In the XML document, Document Type Definition is added to specify the rules of the XML document. In the case of elements, its subelement and its order, whether it is an arbitrary choice ('?'), whether to display more (general meaning '*' or '+'), whether sub-elements can replace each other ('|') and the data content type are specified. $\mathcal{L}_{\varepsilon t}$ is a set of DTD element identifiers and is a set of character strings that obtained "Label*" by linking names in the sign and label of {*, +, ?}.

Definition 1 Document Type Definition

- Tuple $t = (V_t, \bar{v}_t, E_t, ØE_t)$
- $V_t = V_t^e \cup V_t^a$ (elements and attributes)
- $\bar{v}_t$ is a node representing all DTD elements
- $E_t \subseteq V_t \times V_t$ is set of edges($e \in E_t$ is element-subelement)
- $ØE_t : E_t \rightarrow \mathit{Label}^* \cup$ {union, content}(edge labeling function)

XACL [7,8] consists of two parts. The access evaluation finds the appropriate policy for the access request and makes access decisions as well as conditional execution. The request execution updates the target data appropriately or provides data to access requester if the access decision is "allowed". In this study, it is not possible to define attributes in detail such as large-capacity XML data and policy layer is also problematic. Also, XPath language is not used completely.

XrML [9] is a method of using digital signature, and pattern matching through XPath is also possible. However, when considering secure security, problems arise when accessing large capacity XML data. There is a problem in dealing with resources such as large capacity XML data that can be modified in the Web environment.

[10] expressed XML data as a tree. The subject layer is shown in the XML subject sheet. This model cannot safely protect all kinds of nodes. Moreover, this study suggests behavior that is not just language.

The hole-filler proposed a technique to perform query processing by expressing the relationship between pieces [11–16]. XFrag proposed XFPro which improved query processing time [17]. These studies only discussed query processing techniques for streams in the client. However, problems such as processing and memory waste arise due to the additional information.

[18] has no possibility to protect elements in a mobile computing environment. Furthermore, the access to large capacity XML data provides only read operation. In addition, applying a labeling technique that denies access to elements limits the utilization of data because subelements of rejected elements also deny access [16,19].

3 Resource Efficient Secure Access Policy (RESAP)

3.1 Semantics of Large XML Data Subject Policy

Security technologies related to mobile computing include user authentication, data protection, and security protocols. However, this study focuses on large-capacity XML data access. Figure 2 illustrates the secure access policy process for XML data. The authority information is information about objects that the subject can or cannot access. The security policy defines the principle of allowing or denying access. The condition or method of allowing access includes the partial permission of the user's access request and the filtering of unauthorized data. An authorization policy is a representation of security for very small units of XML data. A very small unit of authorization policy is used to authorize a user or object. It is also used when multiple users' rights conflicts occur with respect to an object. The propagation policy is used as a method to determine the priority among rights when conflicts occur, and to set rights when different rights are set on the same element [12,15,16,20,21].

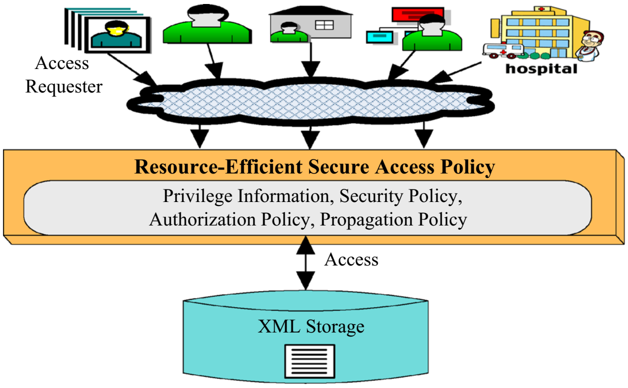

Access Requester
hospital
Resource-Efficient Secure Access Policy
Privilege Information, Security Policy, Authorization Policy, Propagation Policy
Access
XML Storage

Fig. 2 Secure access policy in mobile environment

In general, a subject can refer to an identification number or a location from which a request is made. The location is associated with a numeric IP address. The User_ID means the User_ID of the server to which the user has connected. It supports user groups and location patterns to allow very small authorization specifications that can be applied to users and devices.

The large-capacity XML data authority type is defined as the authority for XML and the authority for elements in XML document. The XML top-level element can only have one, but the sub-element is composed of several layers. The authorization for XML is as the same as Definition 2.

Definition 2 Authority for large capacity XML document and elements

- DR(Define Read): Read the definition part of the XML document
- IR(Instance Read): Read Instance documents
- IG(Instance Create): Create instance document
- IW(Instance Write): Instance documents (read, create, modify)
- Element Read(ER): Read element data
- Element Write(EW): Read and modify element data

The authority for instance documents exists in a mutually combined form and is expressed in a layer. Figure 3 below shows the authority of the instance document as a layer structure. Figure 3 show that the IW authority also has IG and IR authority.

The authority exists in a form that is inter-coupled with instance authority and this can be expressed as a layer structure. Figure 4 shows the layer structure of instance and element authority. The parent authority has authority to the child authority.

3.2 Dynamic XML Data Object Policy

URIs indicate resources that need to protect large capacity XML data [22]. Since a standard language is introduced,

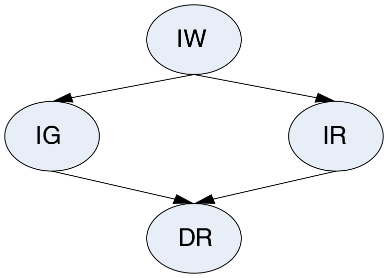


Fig. 3 Permission on XML documents

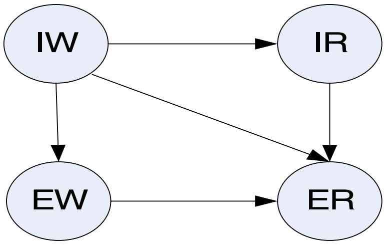


Fig. 4 Hierarchy of element privilege type

users can use the syntax and semantics of the language well. XPath also provides many functions that can use character strings, numbers, and node operations. XPath is a list of element names or predefined functions distinguished by the slash(/) in the tree structure of the document. The tree represents all attributes or elements named XPath $l_1/l_2/\cdots/l_n$.

In this paper, the browsing authority in the mobile environment allows the subject to read information in the element or retrieve information along the link. The read authority grants the subject the authority to view elements and components and search authority is the right to identify the existence of a particular link and all the links in that element and search according to it. Distinguishing between read and search authority allows a subject to access an element while not displaying the link between elements and several elements. Write authority allows the subject to modify(delete) the element content or add new information to the element. The add authority does not delete the existing information, but the subject writes information in the element or includes a link to the element. In contrast, write authority allows a subject to modify element content and include links in the element. If the subject has the write authority for an element, it can also delete the element. In this study, the secure access policy of large-capacity XML data consists of <subject, object, action, ALTG(action label type group), sign, type>.

Definition 3 secure access policy

- Subject ∈ role (AS: Authorization Subject) is User_ ID(Password), IP, Certificate (Certificate_Password)
- Object ∈ authorized object XPath 1.0 (XML element)
- Action: Read, Write, Create, Delete
- ALTG: Read Label Group: A set of operations(Read)
- DSALG: A set of operations that alter and manipulate structure (Write, Delete, Create))
- Sign ∈ {+, −} is authority (Permission, Denial)
- Type ∈ {L, R, LDH, RDH, LD, RD} is the attribute value of the authority

In a secure access policy, authorization can be specified in large capacity XML document or DTD. Detailed authorization to DTD is applied to large XML data that is an instance of DTD. The department supports detailed authorization in relation to DTD, and specific sites can support authorization in detail with respect to individual documents as well as DTD. An action is operations that a subject can perform. ALTG determines to what extent large capacity XML data is allowed. A code is a representation of the permission and rejection of authority, and Type means an attribute value of authority. DTD is propagated to the DTD instance due to the propagation relationship. A secure access policy provides various protection levels to valid data instance of a DTD if the protection level of DTD is very small. Data protection is carried out in detail on various access policies for DTD according to the protection to be implemented using propagation. Authority conflicts are supported using DTD-level authorization to determine priorities. When assigning priorities between authorization, describe the highest priority first.

4 Access comparison for managing large XML data

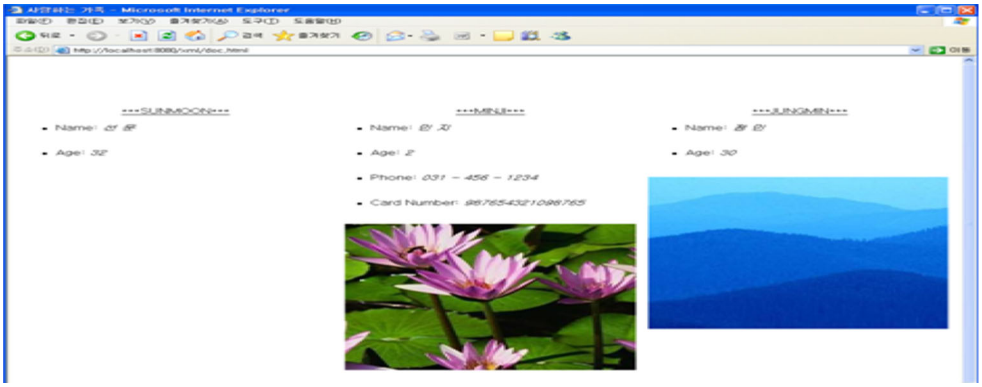
4.1 System Comparison

This paper made some suggestions about secure access policy techniques for large capacity XML data. The characteristics of the proposed policy are compared with the existing ones. The criteria of comparison include whether to satisfy the secure security requirements of large-capacity XML data, access subject, level of protection of subject unit, access objects, content based, propagation, authorization, and update methods. A comparison between the proposals for these requirements is summarized in Table 1.

Since the security requirements for XML data may include each element with a different security level of the XML data, the various and secure layers of security should be supported. RESAP proposed in this paper supports very small unit access policies in very small units by defining the accessible area of the target document as the element unit. The common disadvantage of the existing studies is to define the subject of access to XML data as an individual user. By defining and applying the access rules in a form that specifies the relationship between the user and the object to be accessed one-to-one, one-to-one access is possible between the user and the object to be accessed, but is not applicable in an environment with a large number of users or large capacity data and accompanying instance documents. RESAP performs access to part of the document as well as the entire document by analyzing the large-capacity XML data structure to be accessed and allowing each element to perform a secure access policy. The content-based approach(applying authority differently according to specific areas) was supported only by Damiani and RESAP. All of the above methods use the layer structure of large capacity XML data that supports policy propagation. As a result, except for RESAP, none of the

Table 1 Access system comparison

Requirement	Damiani	Gabillon	Hada	RESAP
Access Subject	Subject and IP Group	Subject Group	Subject Id Role and Group	Subject and user_ID, IP Group Qualification
Subject (unit protection)	XPath element, Attribute	XPath element, Attribute	XPath element	XPath element, Attribute
Access object	Instance document	Instance document element	element	Instance document element
Content base access	Yes	No	No	Yes
Propagation	Yes	Subtree Policy	Yes	Yes
Authority	Read	Read	Read, Create, Delete	Browser, Insert, Delete, Modify
Update	No	No	No	Yes


사랑하는 가족 - Microsoft Internet Explorer

http://localhost:8080/xml/doc.html

SUNMOON

- Name: 선 문
- Age: 32

MINJI

- Name: 민 지
- Age: 2
- Phone: 031 – 456 – 1234
- Card Number: 987654321098765

JUNGMIN

- Name: 정 민
- Age: 30

Fig. 5 Authorization result

existing ones considers efficient secure access policies based on various data strategies.

4.2 RESAP Evaluation

In order to implement the policy proposed, the prototype of RESAP was designed using Apache XML Parser, Internet Explorer 10, Java 8 in CPU Intel Core2 3.0 GHz, hard disk 520 GB, memory 4.0 GB, Windows 7 operating system. When the authenticated accessor requests the resource, RESAP checks the user's authority and determines whether or not to provide the requested data.

The access method for XML data has been proposed in recent years. The system is very similar to the previous one for object-oriented database. In particular, it does not consider cases not conforming or partially conforming to DTD. Therefore, existing methods do not support secure security manager of large capacity XML data.

Figure 5 is an example where minji logs in and shows the results according to the authorization settings. Minji can not see the phone number(authorization sign value = "–" /). Also, it can not see card information. However, if the authorization associated with minji is as follows, minji has the authority to view all information related to it, so it can see the whole information.

<subject > minji| ∗ | ∗ </subject>,
<object > /people/person[./name = "minji″] < /object>,
<altg value =″ rlg″/>,
<action value = "read″/>,
<sign value = " +″ />,
<type value = "RDH″/>

The performance evaluation compares the large capacity XML data access for the existing research and RESAP. Performance evaluation of query processing and memory usage was performed using XML document (113.78 MB) and DTD from XMark Benchmark Project [23]. As shown in Fig. 6, RESAP and Hada are configured for the user in four execution modes. Hada includes access information as well as information in the data node according to the data. In RESAP, however, only one change node access information of a document is stored using an access policy for XML data encoding. Therefore, it occurs less than the number of nodes in Hada.

This experiment measured the processing time and memory usage during the query processing, and described all four queries in Fig 7. Figure 8 is a graph showing the time taken for

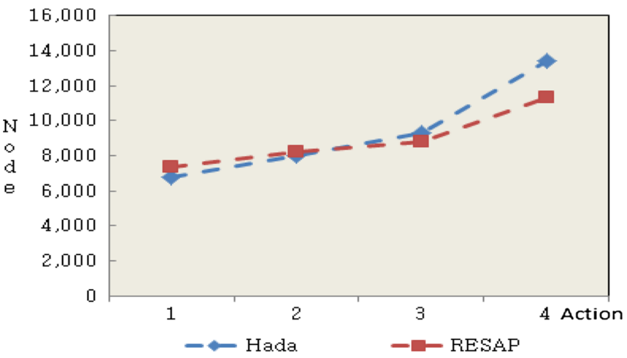


Fig. 6 XML node evaluation

Query	XPath expression
Q1	/site/people/person/homepage
Q2	/site//increase
Q3	/site/emailaddress
Q4	/site/people/person/phone(Delete)

Fig. 7 Query example

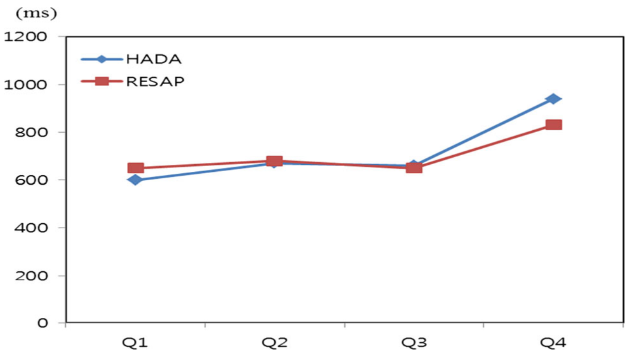


Fig. 8 Processing time

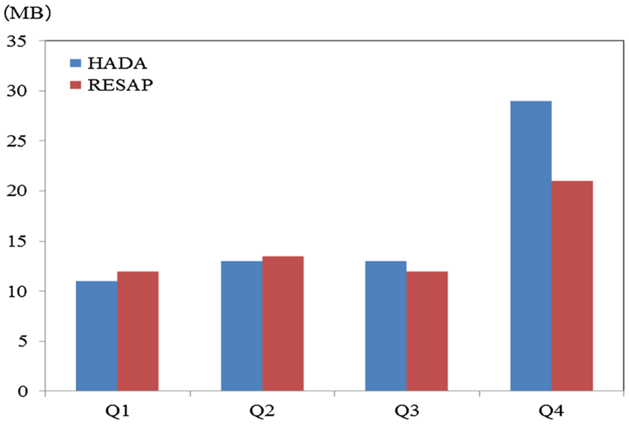


Fig. 9 Memory usage

query processing. Experimental results show that RESAP of this paper is similar to Q1, Q2, and Q3 compared to HADA, and shows excellent performance for Q4 query processing time. Figure 9 is a graph showing memory usage by query. RESAP uses less memory for Q4 than HADA. On the other hand, it shows almost similar memory usage for other queries.

5 Conclusions

This paper proposed a mechanism for secure access policy and data management for dynamic XML data access in a mobile environment. A secure access policy provides protection for very small units of large capacity XML data elements. When looking at both user IDs and groups, a variety of protection requirements can be easily supported while not only providing permission and denial of access but supporting secure access policies as exceptions. In this paper, a policy grants the authority to general users and objects. It is also used when there are conflicts of authority among several subjects on the same object. The RESAP of this study provides many options to efficiently manage large capacity XML data. The existing system does not provide a unique access mode for XML data. This study provides an efficient and secure access for browsing and users in a mobile environment. Based on this, the administrator granted the user the authority to read specific information in the element, search along the link, add, modify and delete element links. Future research should be based on existing research to improve the safe access policy system more efficiently. In addition, it is necessary to study the application of XML data to various applications using large capacity XML data.

Acknowledgements This work was supported by the research grant of Pai Chai Unviersity in 2017.

References

1. Kim, J.C., Chung, K.Y.: Depression index service using knowledge based crowdsourcing in smart health. Wirel. Pers. Commun. **93**(1), 255–268 (2017)
2. Charles P.P., Shari L.P.: Security in Computing, 3rd edn. Prentice Hall, Upper Saddle River (2003)
3. Bishop, M.: Computer Security. Addison Wesley, Boston (2003)
4. Bartel, M., Boyer, J., Fox, B., LaMacchia, Brian , Simon, E.: XML Signature syntax and processing. http://www.w3.org/TR/xmldsig-core/ (2002)
5. Deutsch, A., Fernandez, M., Florescu, D., Levy, A., Suciu, D.: A query language for XML. In: International Conference on World Wide Web. http://www8.org/, (1999)
6. Mohan, S., Sengupta, A., Wu, Y.: A Framework for access control for XML. ACM Journal, vol. v, No. pp.1–38, July (2006)
7. Kudo, M., Hada, S.: XML Document Security based on provisional authorization. In: Proceedings of the 7th ACM Conference on Computer and Communications Security, Athens, Greece, pp. 87–96, Nov (2000)

8. Hada, S., Kudo, M.: XML access control language: provisional authorization for XML Documents. http://www.trl.ibm.com/projects/xml/xss4j/docs/xacl-spec.html, pp. 1–28, April (2002)
9. Guard, C.: eXtenible Rights Markup Language (XrML) 2.0. http://www.xrml.org (2001)
10. Gabillon, A., Bruno, E.: Regulating access to XML documents. In: Proceedings of the Fifteenth Annual IFIP WG 11.3 Working Conference on Database Security, Niagara on the Lake, Ontario, July (2001)
11. Bose, S., Fegaras, L.: XFrag: a query processing framework for fragmented XML Data. In: Proceedings of the WebDB, pp. 97—102 (2005)
12. Kim, J.M., Jung, H., Kang, M.A., Chung, K.: 3D human-gesture interface for fighting games using motion recognition sensor. Wirel. Pers. Commun. **89**(3), 927–940 (2016)
13. Kim, J.C., Jung, H., Chung, K.Y.: Mining based urban climate disaster index service according to potential risk. Wirel. Pers. Commun. **89**(3), 1009–1025 (2016)
14. Chung, Park, R.C.: P2P cloud network services for iot based disaster situations information. Peer-to-Peer Netw. Appl. **9**(3), 566–577 (2016)
15. Jung, H., Chung, K.Y.: P2P context awareness based sensibility design recommendation using color and bio-signal analysis. Peer-to-Peer Netw. Appl. **9**(3), 546–557 (2016)
16. Chung, K.Y., Park, R.C.: PHR open platform based smart health service using distributed object group framework. Clust. Comput. **19**(1), 505–517 (2016)
17. Huo, H., Wang, G., Hui, X., Zhou, R., Ning, B., Xiao, C.: efficient query processing for streamed xml fragments. Lecture Notes in Computer Science(LNCS), vol. 3882, DASFAA, pp. 468–482 (2006)
18. Damiani, E., Vimercati, S., Paraboschi, S., Samarati, P.: Design and implementation of an access control processor for xml documents. In: Proceedings of the 9th International WWW Conference, Amsterdam, pp. 59–75, May (2000)
19. Jung, H., Chung, K.Y.: PHR based life health index mobile service using decision support model. Wirel. Pers. Commun. **86**(1), 315–332 (2016)
20. Yoo, H., Chung, K.Y.: PHR based diabetes index service model using life behavior analysis. Wirel. Pers. Commun. **93**(1), 161–174 (2017)
21. Jung, H., Yoo, H., Chung, K.Y.: Associative context mining for ontology-driven hidden knowledge discovery. Clust. Comput. **19**(4), 2261–2271 (2016)
22. World Wide Web Consortium (W3C) XML path language (XPath) 2.0. http://www.w3.org/TR/xpath20 (2001)
23. The XML benchmark project. http://www.xml-benchmark.org