The Process of Mapping XML Data to Relational Databases from MIS Perspective

Dr. Mohammed Alsaigh
King Abdulaziz University
Jeddah
Saudi Arabia

Abstract

XML became the standard for exchanging information over the web, and massive data is used, modified, transferred and stored continuously using XML files. On the other hand, there are huge existing applications and infrastructure which are created for RDBs. Therefore, there is a need for creating processes for mapping this XML data to the Relational databases (RDBs) in order to be used by the existing applications. In this paper, we investigate the literature to extract the factors that may affect the mapping process, and create a model that represent the relationships between these factors from Management Information Systems (MIS) perspective. Then, we combined two existing algorithms proposed by Chaudhuri et al. (2005) and Wang et al. (2009) in order to come up with a new algorithm that considers all extracted factors.

1. Introduction

As a consequence of the growth of the World Wide Web (Web) and wireless information environments, the applications in information systems are not standalone any more but integrated and shared. Extensible Markup Language (XML) has become the standard for internet data representation and exchange (Chaudhuri et al. 2005;Seng 2003).Collins et al. (2002) state that "Businesses are racing to develop applications that utilize XML for business to business (B2B) transactions and researchers are developing new and innovative methodologies to employ and enhance the features of XML". Thereby, there are chances for growth in the direction of creating efficient and reliable XML storage (Chen et al. 2003; Zhang et al. 2004). In fact, the literature shows two approached for using XML files in transferring data and information on the web. The first approach is to create new XML systems designed specifically for storing XML information. However, this approach will be costly because it requires creating systems from scratch with new technologies (Chaudhuri et al. 2005, Seng et al. 2003). The second approach is to utilize the existence of the relational databases, which more than 20 years of work invested, as repository for XML data. Wang et al.(2009) states that "It seems more attractive to store XML data in existing commercial relational databases to reuse the sophisticated storage and query capability of existing relational databases". In this paper, we will focus on the second way.

In the literature of mapping XML files to RDBs, numerous algorithms have been proposed to achieve this task (Seng et al. 2003).The main issue with these algorithms that there are several factors which are involved in the process of transforming data from XML in RDB, and none of these algorithms consider all of these factors. Furthermore, the development algorithms of mapping XML data to RDBs as any other software development have to go through the development process in order to be implemented correctly, and gathering the requirements is the first step in this process (Satzinger 2003). Therefore, in order to implement an algorithm for mapping XML data into RDBs, it is necessary to identify and consider all factors that involved in the process of mapping the data into the database. In other words, in order to create a good algorithm, you have to have the right model.

In this paper, we will investigate the literature to extract the factors that affect the process of mapping XML data into RDBs in order to improve the performance of process. We will not start from scratch but we will use one of the proposed algorithms and build upon it. In the next section we will review the related work. Next we will explain the methodology and the solution proposed in this paper. Finally, we will conclude the contribution of this paper.

2. Related work

In 2005, Chaudhuri et al. proposed an algorithm for mapping XML data into RDBs which have addressed several factors such as considering the logical design and physical design of the database in the mapping process.

They state that the past work did not consider the importance of physical design of the databases, and its influence on the mapping process. Also they stated that “if the physical design space is not considered, then the logical design space gets inappropriately defined”. In addition, they have proven by giving several examples that there is a need for solving logical and physical designs dependently in order to improve performance. They used greedy algorithm with considering both the logical and physical design of the database for the mapping process. However, the algorithm did not consider other factors such as Information security in XML files, controlling the concurrency while inserting data into database and the impact of the characteristics of the applications used for the mapping process. although that the main purpose of the algorithm which proposed by Chaudhuri et al. is to consider the physical design of the database, the authors did not mention the impact of the physical database partitions on the performance of the process of shredding the data into RDBs. This sub-factor is addressed by Cornell and YU (1990); they show how to formulate the partitioning problem, identify the relevant input parameters, and set up the criterion for partitioning.

On the other hand, Wang et al. (2009) proposed an algorithm for mapping XML data into RDBs. The algorithm addressed important factors which are the considering the application characteristics, query workload, and data statistics. In this algorithm, the authors tried to use the genetic algorithm to find optimal XML to relational mappings. They explained their motivation by stating that “Genetic algorithm has been successfully applied to many applications to solve optimization and learning problems”. According to them the basic concept of genetic algorithms is to represent an intelligent exploitation of a random search within a defined search space to solve a problem. However, the algorithm did not address other factors such as considering the physical design of the database as well as the logical design, and the security of the data of the XML file.

Collins et al. (2002) and Madiraju et al. (2006) proposed an algorithm for mapping XML data into RDBs, and they mentioned that overcoming the semantic heterogeneity of the schema to be integrated in the algorithm is a crucial point of schema integration. That can be achieved by providing users a single uniform interface to facilitate data integration and querying without changing the underlying data sources. The motivation for adding this factor is the intent is to integrate data from different databases in a business environment, and minimizing the impact of having different naming methodology in the databases. However, the algorithm did not consider other issues such as Information security in XML files, and the impact of periodical expansions or modifications in database structure in case of the database requires that.

Data Concurrency is used to ensure the consistency between data source and replicated data. In the process of mapping XML data to RDBs, it is important to assure the consistency between XML files and RDBs. In fact, there is a chance for inconsistency between them as a result for the frequent updates. Fekete and Shasha(2005) addressed the isolation level known as Snapshot Isolation (SI). They stated that “SIis a multi-version concurrency control mechanism that prevents many classical serialization anomalies, but allows nonserializable executions in certain cases”. According to them, SI is widely used due to its good performance, especially the fact that it does not block reads; some major vendors support only SI rather than traditional serializability.

Kunduand Bertino (2008) propose an approach to content dissemination that exploits the structural properties of an XML document object model in order to provide an efficient dissemination and at the same time assuring content integrity and confidentiality. This approach is based on the notion of encrypted post order numbers that support the integrity and confidentiality requirements of XML content as well as facilitate efficient identification, extraction, and distribution of selected content portions. By using such notion, the authors develop a structure based routing scheme that prevents information leaks in the XML data dissemination, and assures that content is delivered to users according to the access control policies, that is, policies specifying which users can receive which portions of the contents. The proposed dissemination approach further enhances such structure based, policy-based routing by combining it with multicast in order to achieve high efficiency in terms of bandwidth usage and speed of data delivery, thereby enhancing scalability.

Huesemann (2002) introduces conversion rules to derive a relational database model from XML schemas. The rules are applied for the design of a database for the management of humanitarian development projects. He states that there is a problem for stakeholders that want to share information with the relational database which is the need for an application that contains information about their projects. He listed five rules that have been used to develop mapping algorithms. They can be used to convert one schema into another one by a program, but the results usually need human revision, especially when complex structures are involved. The rules are:

- For each complex element type, create a table and a primary key column.
- For each element type with mixed content, create a separate table in which to store the PCDATA, linked to the parent table through the parent table's primary key.
- For each single-valued attribute of that element type, and for each singly-occurring simple child element, create a column in that table. If the child element type or attribute is optional, make the column null-able.
- For each multi-valued attribute and for each multiply-occurring simple child element, create a separate table to store values, linked to the parent table through the parent table's primary key.
- For each complex child element, link the parent element type's table to the child element type's table with the parent table's primary key.

Furthermore, Guynes and Vanecek have differentiated between Data Administrator (DA) and Database Administrator (DBA) activities. This differentiation would help in determining the responsibilities of the activities of the mapping process. Guynes and Vanecek report the results of a field survey conducted to determine the critical success factors (CSFs) in data management. Authors have offered different views of the objectives and functions of data management and have offered various definitions of database management, data administration, and database administration.

3. Methodology

Algorithms are computer program in their nature (Giudice and Fargione 2007). Therefore, building an algorithm should go through the same process of creating any program. In fact, there are four major stages in creating any program: 1) Requirements gathering. 2) Analysis. 3) Design. 4) Implementation and test (Satzinger 2003). Transferring from stage to another requires specific documentations to transfer the knowledge to the other team such as the Unified Modeling Language (UML) diagrams (Satzinger 2003).

Therefore, in this research we will start the process of building the algorithm of mapping the XML files to RDBs by extracting the factors that influence the mapping process from the literature (Requirements gathering Stage). The next step is to build a model that shows these factors and the interrelationships between them (Analysis Stage). This will build a comprehensive understanding of the mapping process. Next, we will start the analysis stage by identifying the stakeholders of the process, and determining the sequence of the process. The output of this stage will be delivered to the design stage, and we will use UML diagrams for that purpose. The results of this investigation will be used to improve an existed algorithm which is proposed by Chaudhuri et al. (2005) by combining it with another algorithm proposed by Wang (2009) in order to come up with a new algorithm considering all factors. Finally, the contribution of the proposed model and the proposed improvement to the algorithms will be evaluated.

4. The Model of the Mapping Process

As it is explained in the methodology of this paper, in order to build the model, the following tasks should be achieved: 1) Extract the requirements from the literature. 2) Identify the stakeholders. 3) Determine the sequence of the process. 4) Create UML diagrams to represent the process.

4.1 Extract the Requirements from the Literature

As we have seen in the related work section in this paper that there are several factors should be considered in order to determine the requirements of the mapping process. These factors are summarized in the following list:

a. Considering the logical designs of the database as well as the Physical design because this will improve the performance of the mapping process as proven by Chaudhuri et al. (2005).
b. Concurrency: because XML data is expected to be inserted in to RDB in frequent bases, it is important to control the concurrency in order to manage the process of inserting the data from different XML files into RDB while another process is updating, reading, or deleting data.
c. Security: security is an essential factor in the process of transferring data from XML files to RDB. This is because this process became very common, and many researches consider XML as a standard tool for transferring data over the web.
d. Application characteristics, query workload, and data statistics should be considered in the process of mapping XML data to RDBs.
e. Integrating the semantic heterogeneity of the schema

4.2 Identify the Stakeholders

Identifying the stakeholders and their responsibilities in the process of mapping XML data to RDBs is an important step in building the model. Unclear or mixed responsibilities in any project may lead to conflicts, doing tasks more than one time, or skipping some essential tasks. In this paper, we used Data Administrator (DA) and Database Administrator (DBA) activities listed by Guynes (1996) to identify the responsible person for each factor listed in previous part of the solution. Table 1 shows these factors, the responsible person, and the original responsibility that covers this factor according to Guynes categorization. A third actor will be used in the model is the creator of the XML files.

Table1: Responsibilities in the Mapping Process

	Factor	Res	Original Responsibility
1	logical designs	DA	Developing logical data models
	Physical design	DBA	Determining physical descriptions
2	Concurrency	DBA	Specifying database access policy
3	Security	DBA	Selecting and designing security techniques
4	Application characteristics, query workload, and data statistics	DBA	Designing integrity controls for database applications.
5	Semantic integration	DA	Enforcing data dictionary standards

4.3 Determine the Sequence of the Process and the Interrelationships between Factors

After identifying the requirements and the stakeholders in the process, we need to understand the sequence of the process in order to implement the process of mapping XML data to RDBs. The process starts by the person who creates the XML files, and she or he has to arrange with the DBA for the encryption method that will be used in the process in case it is needed. Next step, the DBA will decrypt the data in the XML files when he received them. This shows why arrangement is required between XML creator and DBA. Next, DA will match the names between attributes in XML and DB tables. At the same time, the DBA will assure the compatibility between the application characteristics and the data in the XML files. This may include the ranges of the variables, size of data, and etc. Then, DA and DBA will work together on using the physical and logical design of the database to get the best possible mapping. Finally, DBA will monitor concurrency of the database during each process. Figure 1 shows the proposed model that represents this process.

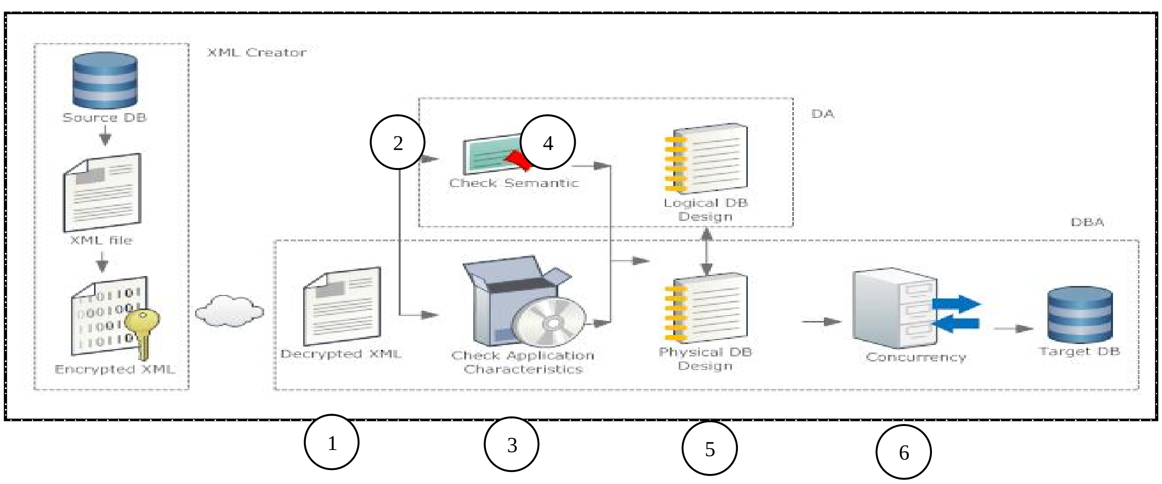

Figure 1: The proposed model for Mapping Process

4.4 Create UML Diagrams That Represent the Process of Mapping XML Files to Rdbs

In order to continue on developing the mapping process, the analyses that have been done in this paper should be documented for the next step which is out of the scope of this paper. We chose UML for documenting the analyses because of its popularity in the field of systems development. Three use cases in addition to one sequence diagram are used to document the analyses that done in the previous part. See figures 2-3.

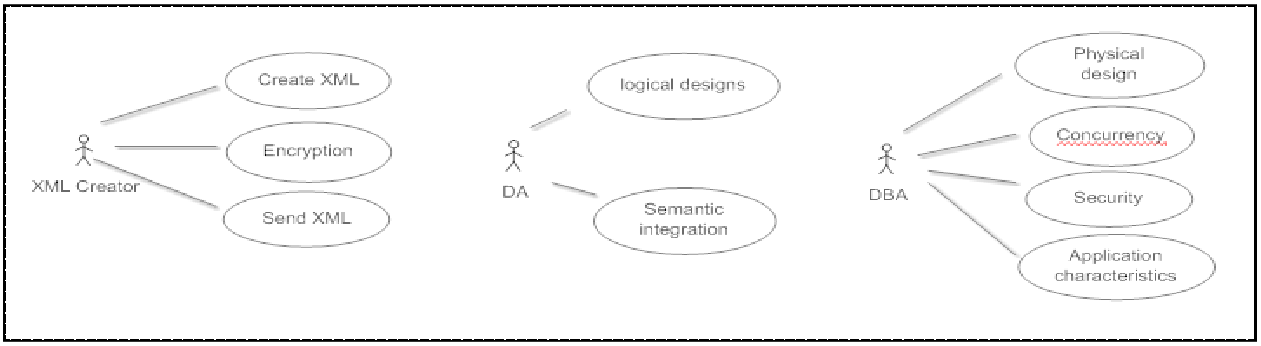

Figure 2: Actors' use cases in the Mapping process

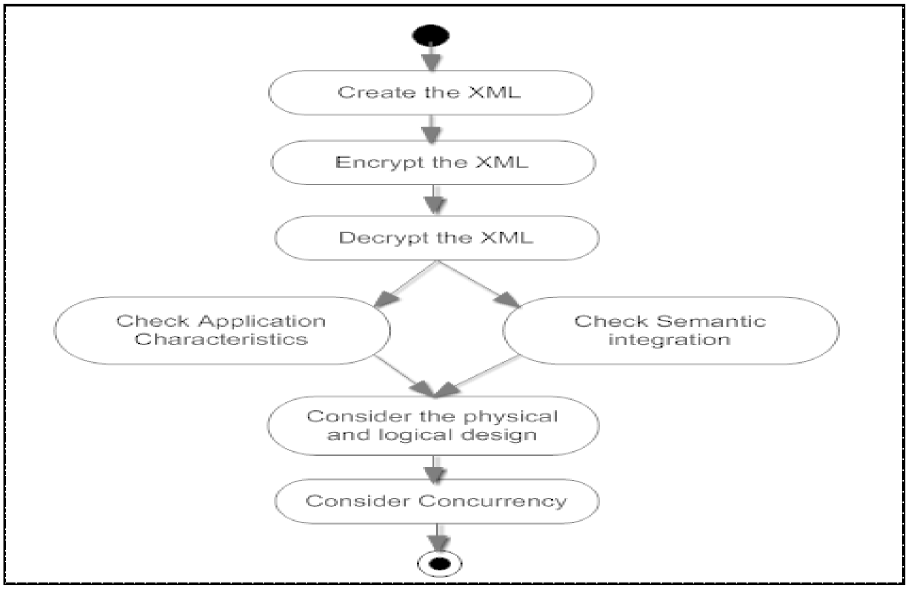

Figure 3: The Sequence Diagram of the Process

5. Improve Existing Algorithms

In this section, we will create a new algorithm by combining two algorithms already produced by other researchers; in addition to using a security component produced by a third researcher. The two algorithms are proposed by Chaudhuri et al. (2005) and Wang et al. (2009), and we will call them algorithm A and B respectively in the rest of this paper. Also, a quick glance has been given about them in section 2 of this paper. Therefore, in order to improve the existing algorithms, we need to go through the stages shown in Figure 1. Also, we will assume that all XML files are accompanying with XSD schema specifications, and X-path queries. This is justified by the popularity of both of XSD schema and X-path queries in the environment of XML. Moreover, to make our solution consistence with other solutions, we will borrow some abbreviations from Chaudhuri et al. (2005) which are listed in table 2.

As we have seen in section 2, algorithm A covers one factor which is considering the physical and logical design of the database by using greedy algorithm. On the contrary, algorithm B covers two factors which are Application characteristics and semantics. On the other hand, we found that Hwang and Chang (2004) produced a component that can be added to the solution to cover the security factor.

The basic idea of our proposed solution is to replace the greedy algorithm used in algorithm A by the genetic algorithm which is used in algorithm B, and add the security component to them. The expected result is having a new algorithm that covers the four mentioned factors. Notice that Concurrency factor is out of the scope of this solution. In the following subsection, we will discuss this combination in details.

Table 2: the Abbreviations That Will be Used in This Paper

T	XML file with XSD schema
W	XPath workload that consists of a set of ðQi; fiÞ
Qi	XPath query and fi is
Fi	The weight associated with Qi.
R	Relational schema
M	Mappings that map T to R

5.1 The Security and Semantic Components

For the security factor, DBA has to decrypt the XML file and its XSD schema (T) which has been encrypted by the senders. For that purpose, we will use an algorithm proposed by Hwang and Chang (2004) with some modifications for simplification reasons. The main idea of this algorithm is to use a tool called Document Security Language (DSL) created by them in order to encrypt and decrypt XML files. Figure 4 explains this idea. According to them, "this algorithm offers a security mechanism which integrates element-wise encryption and temporal-based element-wise digital signatures". In this paper, we will use only the element-wise encryption to maintain the performance and simplicity of the whole framework. Figure 5 shows an example provided by Hwang and Chang (2004) for the element-wise encryption in XML files. The reason of choosing this algorithm is that the DSL tool includes a definition for the ''standard DSL algorithm downloading and linking protocol'' which fulfills automatic algorithm download and linking requirements in the operational model which makes the DSL based securing tool configurable. The basic idea of this algorithm of encrypting an XML document is to employ existing cryptography to encrypt an XML document as a whole. The receiver of an encrypted XML document then decrypts it with the appropriate key and algorithm (Hwang and Chang 2004).

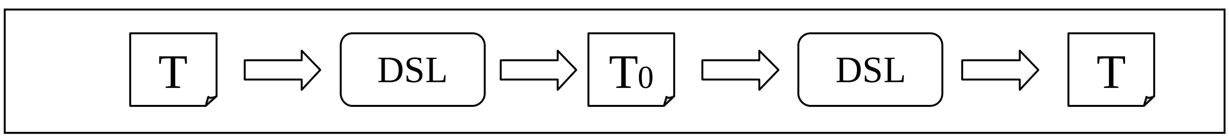

Figure 4The Proposed Encryption Process

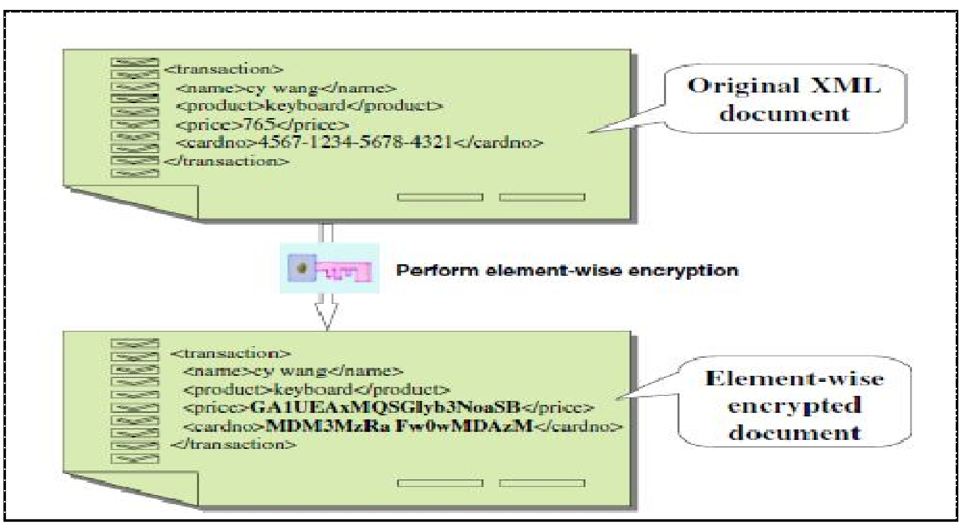

Figure 5 Example of Element-Wise Encryption

For the semantic factor, DA has to check the semantic in the attributes name between the XML files and RDB. Attributes in both sides may have the same names but different meanings. DA is responsible for notifying DBA about these similarities (Madiraju et al. 2006).

5.2 The Physical and Logical Design Consideration

According to Chaudhuri et al. (2005), there are five types of transformations in XSD files, and they can be categorized into two categories: subsumed which are always subsumed by vertical partitioning in the physical design, and non-subsumed as shown in table 3.

Noticing that Subsumed transformations may be combined with non-subsumed transformations which may create a combination that is not covered by physical design, thus the search space in the logical design phase should include the combinations in addition to the non-subsumed transformation; while the subsumed ones will be covered by the physical design. Chaudhuri et al have proven that considering this in the mapping process will enhance the performance significantly. Therefore, the work load will be divided into two groups as it is suggested by Chaudhuri et al. The first group will contain the non-subsumed transformations as well as the transformations that are combinations from subsumed and non-subsumed transformations. The second group will contain the subsumed transformations only.

Table 3: Transformations Types

	Transformation	Category
1	Outlining and In-lining	subsumed
2	Type Split/Merge	non-subsumed
3	Union Distribution/Factorization	non-subsumed
4	Repetition Split/Merge	non-subsumed
5	Associativity and Commutativity	subsumed

5.3 The Application Characteristics Factor

According to Wang et al. (2009), the greedy algorithm, which is used in algorithm A, does not have the capability of considering the applications characteristics in the mapping process. Therefore, they suggest using the genetic algorithm, and they have proven that their proposed algorithm, algorithm B, has better performance than the others which use greedy algorithm especially for large XML files. In fact they stated that “Genetic algorithms are more robust than greedy algorithms in finding optimal solutions when search space is large”.

In fact, genetic algorithm has been used to solve many problems, and the basic of it is to use intelligent random research in a defined search space. Wang et al. (2009) created an algorithm based on genetic algorithm to generate a relational schema from XML files and the attached DTD files. This algorithm can differentiate between attributes and relations according to the given information in the DTD file.

5.4 The Proposed Algorithm

In our proposed algorithm, we will use algorithm A as a starting point, and replace the greedy algorithm part in it by the genetic algorithm proposed in algorithm B. Figure 6 shows the idea of our proposed model, and figure 7 shows algorithm B.

Input:
- XML XSD➔ T
- XML Query work load ➔ W

Output:
- The mapping file ➔ M

The algorithm:
1. C0= Generate Candidate (T,W)
   C1 = Merge type candidate (C0)
   C2 = Split type candidate (C0)
2. While there are unvisited edges do
3. Use algorithm B to create a relation or attribute in R

Figure 6The proposed algorithm

```
Algorithm 1 Relational Schema Generation Algorithm
Input:
  G is a DTD graph with n edges
  B = b1b2... bn is a length-n bit string
Output:
  A relational schema R;
Description:
R = Φ ; Stack = Φ ; top = 0; i = 1;
while There are unvisited edges do
  if Stack is empty then
    Pick the next root node p and create a relation r for p;
    Stack[0].node = p and Stack[0].relation = r; top=1;
    R = R ∪ {r}
  p = Stack[top - 1].node; r = Stack[top - 1].relation;
  if p has unvisited out-edge then
    e = next unvisited out-edge of p;
    q = the child of p pointed by e;
    if bi==1 then
      if No relation exists for q then
        Create a new relation rnew for q;
        R=R ∪ { rnew }; r = rnew;
    else
      r.Attr_List = r. Attr_List ∪ q. Attr_List;
    i++;
    if e.child has unvisited edges then
      Stack[top].node = e.child;
      Stack[top].relation = r;
      top++;
  else
    top--;
```

Figure 7 Algorithm B

6. Evaluation

In this paper, we have proposed two artifacts. The first one is the proposed model, and it will be evaluated by conducting static analysis to show the contribution of the model from IS prospective. Also, we will construct detailed descriptive scenario about the model to demonstrate its usability from practical prospective. The second artifact is the proposed improvement to the suggested algorithm, and it will be evaluated by conducting descriptive informal argument.

6.1 **Evaluating the Model**

6.1.1 Static Analysis

The model is evaluated by examining the following static qualities in the model:

1. Completeness: the proposed model covers all factors that extracted from the literature.
2. Reasonability: The sequence of the steps in the model looks logical and realistic. Also, the process in the model is compatible with what we extracted from the literature.
3. Complexity: The model is simple and there is no overlap between the steps and tasks.
4. Coherence: The model explained the responsibilities of each part, and it identified the person who is in charge of each task.

6.1.2 Descriptive Scenario

The mapping process is like any software project. Therefore, we can evaluate it by comparing it to the life cycle of the software development. In fact, software development has mainly four stages which are: Gathering the requirements, Analysis, Design, and Implementation. In our model which represented in figure 1, we can see that steps 1-3 which are decryption, check semantics, and check application characteristics represent Gathering the requirements because they all used for collecting the data from XML files. Also, we find that step 4 and 5 can be considered as the analysis stage for the process; while step 6 which is the concurrency control can be considered as the design stage for the process. Finally, dumping the data into RDBs is the implementation of the project. In conclusion, we used this scenario to examine the logic of the model and we found it acceptable.

Also, to evaluate the contribution of our algorithm, we will borrow the case provided in the article written by Huesemann (2002), and evaluate our model accordingly. This article explains challenges that arise when humanitarian organizations want to coordinate their development activities by means of distributed information systems. It focuses on information exchange based on the XML and RDBs. We can see that our model considers all issues raised by the author of the article such as:

- Consider the physical and logical database designs.
- Consider the semantic issue.
- Consider the application characteristics.

Moreover, our model considers issues which are not mentioned by the author such as:

- Consider XML data security.
- Consider concurrency.

6.2 **Evaluating the Added Improvement**

In fact, it is difficult to evaluate the contribution of our proposed algorithm because it is still a concept more than a real algorithm. However, we will evaluate its expected performance by using the results of the evaluations that have been used to evaluate the components of it. Therefore, I will use the evaluation of two components: the usage of the physical and logical deign (algorithm A), and the usage of the genetic algorithm (Algorithm B).
Chaudhuri et al (2005) implemented their proposed algorithm which search the logical and physical design of the database together, and compare it to two other algorithms: 1) a Naive- Greedy algorithm as a straightforward extension of the greedy algorithm proposed in which calls the physical design tool for every enumerated logical mapping and 2) a Two-Step algorithm that first greedily selects the minimal cost logical mapping without considering physical design. The findings of their evaluation are as follow:

- It is important to search logical and physical design together because the algorithm searching them independently leads to query performance on average a factor of two worse than the algorithms searching them together.
- The proposed algorithm drastically reduces the running time of the algorithm (on average by two orders of magnitude) with little or no degradation in terms of quality of results.

On the other hand, Wang et al (2009) have tested their proposed algorithm on seven cases, and they concluded that their algorithm produced better mapping than other algorithms that are using greedy algorithm. Therefore, we can conclude that using genetic algorithm with consideration of the physical and logical design of the database will provide better performance.

Conclusion

XML became the standard for exchanging information over the web. Because of the existing infrastructure and application for RDBs, it was feasible to have tools for transforming information from XML files to RDBs from cost prospective. However, mapping process is needed because of the difference in structure between databases and XML files.

Building a process for mapping XML data to RDBs is as creating any software program; it should go through the well-known four stages: Gathering the requirements, Analysis, Design, and Implementation. By reviewing the literature, we found that most algorithms that created for mapping XML data to RDBs did not consider all factors that impact the process. Therefore, in this paper, we investigated the literature to extract all factors that impact the process, and proposed a model that represent the sequence of the process and the relationships between all factors. In addition, we documented this analysis into UML diagrams to be used in the future for implementing mapping algorithms.

Finally, the proposed model has been used to add some enhancement to two existing models. The enhancement is achieved by combining these two algorithms, and producing one algorithm that considers the factors which are considered by both of them. The new algorithm has been evaluated theoretically because of the available time and resources. Practical evaluation for the algorithm is an opportunity for future research.

References

SurajitChaudhuri, Zhiyuan Chen, Kyuseok Shim, and Yuqing Wu. (2005). Storing XML (with XSD) in SQL Databases: Interplay of Logical and Physical Designs. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 17, NO. 12.

Stefan Huesemann (2002). Information Exchange Between Humanitarian Organizations: Using the XML Schema IDML. Journal of the Association for Information Systems, Volume 3,1-26

Praveen Madiraju, RajshekharSunderaman, Shamkant B. Navathe, and Haibin Wang (2006). Semantic Integrity Constraint Checking for Multiple XML Databases. Journal of Database Management, 17(4), 1-19.

Jia-Lang Seng; Yu Lin; Jessie Wang; Jing Yu (2003). An analytic study of XML database techniques. Industrial Management + Data Systems; 103, 1/2; ABI/INFORM Globalpg. 111

Xiaoling Wang, Jinfeng Luan, and Guimei Liu (2009). Towards optimal workload-aware XML to relational schema mapping. Ann Oper Res 168: 133–150

Samuel Robert Collins, ShamkantNavathe, and Leo Mark (2002). XML schema mappings for heterogeneous database access. Information and Software Technology 44 251-257.

Fabio Giudice, Giovanna Fargione (2007). Disassembly planning of mechanical systems for service and recovery: a genetic algorithms based approach. Journal of Intelligent Manufacturing. London: Jun 2007. Vol. 18, Iss. 3; p. 313.

ALAN FEKETE and DENNIS SHASHA (2005). Making Snapshot Isolation Serializable. ACM Transactions on Database Systems, Vol. 30, No. 2, Pages 492–528.

Ashish Kundu and Elisa Bertino (2008).A New Model for Secure Dissemination of XML Content. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 38

DOUGLAS W. CORNELL AND PHILIP s. Yu, (1990). An Effective Approach to Vertical Partitioning for Physical Design of Relational Databases. 1EEE Transactions on Software Engineering. VOL. 16. NO. 2.

Carl Stephen Guynes and Michael T. Vanecek. (1996). Critical Success Factors in Data Management. C.S. Guynes, M.T. Vanecek / Information & Management 30, 201-209

Chen, Y., Davidson, S. B., Hara, C. S., & Zheng, Y. (2003). Rrxf: Redundancy reducing xml storage in relations. In Proc. of the 29th VLDB conference (pp. 189–200).

Zhang, N., Kacholia, V., &Özsu, M. T. (2004). A succinct physical storage scheme for efficient evaluation of path queries in xml. In Proc. of the 20th ICDE conference (pp. 54–65).

Satzinger, John and Jackson, Robert (2003). Making the Transition from OO Analysis to OO Design with the Unified Process. Communications of the Association for Information Systems Volume12, Pages 659-683.

Hwang, Gwan-Hwan and Chang, Tao-Ku (2004). An operational model and language support for securing XML documents. Computers & Security 23, 498-529