INTEGRATING XML DATA INTO MULTIPLE ROLAP DATA WAREHOUSE SCHEMAS

Soumya Sen[1], Ranak Ghosh[2], Debanjali Paul[3], Nabendu Chaki[4]

[1,4]University of Calcutta Kolkata -700009 West Bengal, India

[1]iamsoumyasen@gmail.com
[4]nabendu@ieee.org

[2,3]Barrackpore Rastraguru Surendranath College, Kolkata - 700120 West Bengal, India

[2]ranakghoshmail@gmail.com
[3]debanjali.rimi@gmail.com

ABSTRACT

Data Warehouse is one of the most common ways for analyzing large data for decision based system. These data are often sourced from online transactional system. The transactional data are represented in different formats. XML is one of the worldwide standards to represent data in web based system. Numbers of organizations use XML for e-commerce and internet based applications. Integration of XML and data warehouse for the innovation of business logic and to enhance decision making has therefore emerged as a demanding area of research interest. This paper focuses on integrating XML data based on multiple related XML schemas, to an equivalent data warehouse schemas based on relational online analytical processing (ROLAP). This work bears a high relevance towards standardizing of the ETL phase (Extraction, Transformation, and Loading) of the OLAP projects. The novelty of the work is that more than one data warehouse schemas could be identified from a single related XML schema and each of them could be categorized as star schema or snowflake schema. Moreover if the individual schemas are found to be related according to the analysis, fact constellation could be identified. A new data structure, Schema Graph has been proposed in the process.

KEYWORDS

XML, Data Warehouse, Fact Constellation Schema, Star Schema, Snowflake Schema, Schema Graph

1. INTRODUCTION

Data warehouse [10] is subject-oriented, non-volatile, integrated, time variant collection of data which helps in developing strategic decisions. The popular way of describing data warehouse is through multi-dimensional model [10]. Multidimensional model is based on some central theme which is represented by fact table [17]. A fact table is constructed from multiple dimension tables [17]. The association between these fact table and dimension tables are generally represented through three data warehouse schemas namely star schema, snowflake schema and fact constellation.

XML is well established standard for semi-structured data and also poses several benefits in web environment. Data could come from different heterogeneous sources [6] and in order to integrate

these heterogeneous data they could be converted to XML. Relational Model on the other hand is the most standard and structured way of representing data model. Numbers of researches have been made over the time to map different data models to relational model. XML is also no exception. Works in [1, 2, 12, 13] show different ways of transformation from XML to relational model schema. In this paper the focus is on converting XML schema to multiple data warehouse schemas based on ROLAP (Relational Online Analytical Processing). Thus we are not concentrating on the techniques of converting XML to relational model. Here we refer to some of the existing methods to transfer XML to data warehouse paradigm. XML data is associated with DTD [11] or XML schema [11]. XML provides Document Type Definition (DTD), which explains precisely what elements could appear as document and what the contents of the elements and attributes are. The approaches [7] [8] [9] show how XML data based on DTD have been converted to data warehouse schema. However DTD have some limitations. DTD do not have any built-in data types; also do not support user-derived data types and allow only limited control over cardinality. XML schemas are more powerful to represent XML document structure and overcome the limitations of XML DTD. XML schema design could be of 2 types namely Russian Doll Design [14] and Salami Slice Design [14]. The Russian Doll design corresponds to having a single global element that nests local elements. The Salami design corresponds to having all of the elements defined within the global namespace and then referencing the elements.

The increasing use of Web makes XML an important source of storing data for its semi-structured nature. These data are often required to be processed analytically by the industrial organizations or corporate for decision support system [15]. Data warehouse allows data to be processed analytically based on OLAP. ROLAP is the most common way of implementing OLAP. Due to these reasons data stored based on XML are required to be converted into data warehouse schema for OLAP processing. As discussed earlier, XML schema is better than XML DTD, we focus on the conversion of XML schemas into ROLAP. Many works have been reported to integrate XML data in data warehouse. [1, 2, 3, 4, 5]. The paper [2] converts XML schema either to star schema or snowflake schema. Moreover this paper works only with single XML schema. The paper [3] proposes a method to design multiple cubes of multidimensional model from XML schema. There has been work to convert the contents of the XML schema to multiple schemas of the multi dimensional model [4]. However all these generated schemas are converted to star schema only. The paper [5] proposes XML schema conversion to OLAP cube by identifying fact and dimension tables.

However, the existing methods studied and cited here are only capable to identify a single data warehouse schema from a given XML schema. These research works do not consider an XML schema which consists of more than one root element (may or may not these root elements are related). As this consideration is missing fact constellation schema could not be identified, because the connections among different root elements are out of knowledge. Thus even if the facts are related they are not being treated within a single data warehouse schema, rather they are represented as discrete facts. The possible sharing of dimension is out of scope. This leads to failure of identifying some of the business processes which are actually related. Thus process integration is not being realized properly where as the business process integration is the framework of ERP [16]. In this paper through our proposed mechanism we focus on these issues. Here a framework is proposed to detect more than one (if exists) data warehouse schema from the given XML schema and also find out whether these multiple data warehouse schemas are related among themselves to form fact constellation. In this paper we consider Russian Doll XML schema, because every root element prompted to a fact and the nested elements are considered to be its dimensions.

This work is an extension of [1] in which a new mechanism has been proposed to obtain the data warehouse schema from related XML schema. In [1] at first XML schema is converted to a schema graph (described in Section 3.2). Schema graph is a new data structure proposed here for

the conversion process. In next stage the fact and dimension tables are being identified from the schema graph (described in Section 3.3). Once these tables have been identified, the measure of fact table is taken from the user. In the next section (Section 3.4) depending on the relationship between dimension tables and fact tables one or more star schemas and snowflake schemas are being identified. Based on the nature of relationship among these schemas fact constellation is formed. The extension work in this paper is aimed to enhance the ETL (Extraction, Transformation and Loading) [18] [19] phase of data warehouse projects. Data could be extracted from the XML schema according to the proposed methodology and these data are transformed accordingly for the loading purpose in the data warehouse schema. Once the proposed methodology runs on the ETL phase of existing XML schema the data warehouse schema is identified. If the schema structure of the XML changes then the warehouse needs to be modified to reflect the change in source data, hence the extraction, transformation and loading need to be performed again.

2. PROPOSED DATA STRUCTURE

Schema Graph: A schema graph is a representation of entities found in the XML Schema. The graph has the following properties:

a. It is a level wise separable graph.
b. The entities encountered in the XML are represented through vertices of the graph.
c. The name of the vertices of the Graph would be same as the name of the entities.

Holder Element (HE): The elements that have no predecessor are in the Schema Graph are called Holder Elements. They are placed in Level-1 of the graph

Contained Element (CE): The elements that are directly connected to the HE's are called Contained Element. They are placed in Level-2 of the graph.

Secondary Elements (SE): The elements that are directly connected to the CE's are called Secondary Elements. They are placed in Level-3 of the graph.

If elements in the graph appear as connected to SE, they would be placed in level-4. The new vertices that would be connected to the vertices of level-4 would be placed in level-5. Subsequently new level could be created if the new entities appear in the graph connected to the previous level. All the entities are represented through rectangular vertex in Schema Graph. The attributes are represented in the Schema Graph in an Oval shaped vertex. They are connected to the corresponding entities. In Figure 1 entities are shown only.

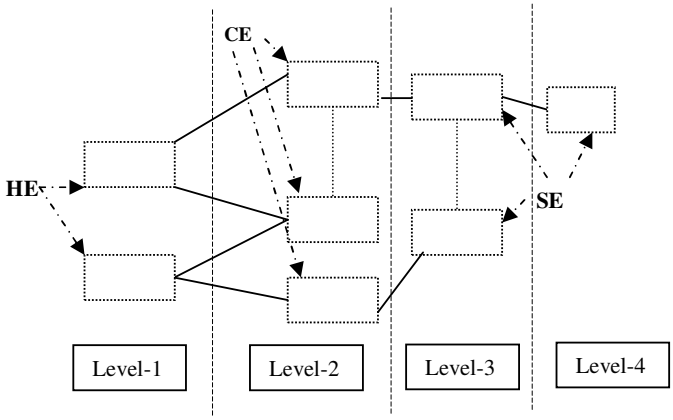


Figure 1. Schema Graph Along with HE, CE and SE

3. CONVERSION OF XML SCHEMA TO WAREHOUSE SCHEMA

3.1. Overview of the Proposed Framework

The modeling of Warehouse Schema from a related XML Schema is performed in two steps. At first Schema graph is formed from the XML schema (described in section 3.2). The elements of the schema graph are identified as HE, CE and SE. In the next stage fact tables and dimension tables are identified. Moreover if any element doesn't have any primary key then a key attribute is added to it (described in section 3.3). Each HE would correspond to a fact table and would make an entry in the fact table, the key attribute of the CE's that are connected to the HE are entered in the corresponding fact table for that HE. CE's would be the dimension tables. If SE's are found connected with CE the primary keys of SE's are placed in CE. If SE's are present even after level-3, the primary keys of the higher level are placed in the table corresponding to the SE of immediate lower level. The warehouse schema structure is identified in section 3.4). There are three separate methodologies for three data warehouses schemas. Identification of star schema, snowflake schema and fact constellation are described in section 3.4.1 through 3.4.3.

3.2. Procedure to Build Schema Graph from Related XML Schema

a) Find out those entities in XML schema that have no predecessor and denote them as the starting vertices or holders for the entire graph. These entities would be known as **HE**. They would placed in the Level-1 of the graph.
b) For all **HE** (i=1 to n) perform following : (n is the total number of **HE**)
i. Find the sequence of elements under i-th **HE :**
If it is an element then create a vertex for it into the graph and connect it with i-th **HE**. These elements (vertices) would be denoted as **CE. CE** would be placed in the level-2 of the graph.
Else if it is an attribute it would be considered as an attribute of the corresponding **HE**.
ii. For all **CE** (j=1 to m) perform following: (m is the number of CE in the **HE**)
iii. Scan the XML Schema for j-th **CE**:
If it is an element then place it into the graph and connect it with its **CE**. These elements would be known as **SE**. **SE** would be placed in the level-3 of the graph.
Else if it is an attribute place it would be considered as an attribute of **CE**.
iv. For every **SE** (k=1 to p) : (p is the total number of SE at that level)
Repeat the steps to include the entities and attributes as they encountered. Whenever a new entity is added new level is created for it.
End For /* SE */
End For /* CE */
End For /* HE */

Figure 2. Algorithm to Construct Schema Graph from XML Schema

In order to build the schema graph from the XML schema at first the entities are identified. The elements that are not nested within some elements are termed as HE and are placed at level-1 of the schema graph. After that if some element is being found to be declared within the body of HE they are being classified as CE and placed at level-2 of the schema graph. Again if new elements are declared within body of CE they are being placed at level-3 of the schema graph and are termed as SE. If the nested elements are further found for SE they are placed in the next level of the schema graph and are also termed as SE. This scanning would be continued till all the

elements which are being nested are identified. Though new levels are created in the schema graph based on the higher level of nesting all of them are termed as SE, that is, elements in level-3 or more than level-3, all are being termed as SE. This algorithm is shown in Figure 2.

3.3. Identifying Fact and Dimension Tables

While drawing the Schema Graph three entities have been specified: HE, CE, SE. The CE's are chosen as the dimension table and each HE would prompt towards a fact-table. As we consider the Russian Doll XML schema, every root in the schema is the major element which is further detail out through the nested elements. The remaining elements of the schema graph are considered to participate in the warehouse schema in the form of dimension table. If there exists any SE, then the primary keys of them would be included in the corresponding CE's. Similarly, if there are entities beyond level-3, then the primary keys of entities of level (i+1) appear as foreign key in the connected entity of level i. If any entity is found without having a primary key, then primary key is added to it as: Name of entity + "_id"

3.4. Identification of the Schema Structure of Data Warehouse

Once the fact and dimension tables are identified, the corresponding DW schema is to be built using the star schema, snowflake schema and fact constellation schema. This is done by checking the nature of connection among elements within the schema graph. If the schema is found as disconnected, more than one schema is identified. The numbers of distinct components in the schema is same as the number of DW schema identified.

3.4.1 Procedure Star Schema

A data warehouse schema is identified as star schema if the schema graph consists of HE and CE's only. Every fact table is named as the name of HE + "Fact". The primary keys of each of the connected CE are placed in fact table and the CE's are act as dimension tables in the star schema. This algorithm is shown in Figure 3.

Partition the Schema Graph Level wise.
Identify HE
For HE:
a. Form a Fact-Table with the name of HE + "Fact" and Primary key of the HE
b. Specify the CE connected with this HE; include the primary keys of each CE into the Fact-Table.
End For /*HE*/

Figure 3. Algorithm to Construct Star Schema

3.4.2 Procedure Snowflake Schema

Partition the Schema Graph Level wise.
Identify the HE
For each HE:
a. Form a Fact-Table with the name of HE + ”Fact” and primary key of the HE
b. Specify the CE connected with this HE, include the primary keys of each CE into the Fact-Table.
c. For each CE find SEs, if any.
If found Connect it with its CE using the primary key of the SE.
d. For each **SE:**
Check if there are any **SE:**
If further level of SE is found the primary key of the new SE of the immediate higher level is placed in the SE of current level;

Figure 4. Algorithm to Construct Snowflake Schema

A data warehouse schema is identified as snowflake schema if the schema graph consists of HE, CE and SE provided that HE’s are not connected. At least one SE should exist for some CE in the schema graph to be identified as snowflake schema. Every fact table is named as the name of HE + “Fact”. The primary key of the connected CE are placed in fact table and the CEs act as dimension tables. Similarly the dimension tables would contain the primary key of the tables represented by SE. If further levels of SE are found they are represented in the schema by placing the primary key in the immediate previous level of SE tables.

3.4.3 Procedure Fact-Constellation

A data warehouse schema is identified as fact constellation if the fact tables are found connected through the dimension tables. That is if more than one schema shares the same dimension table then those schemas would be combined to form fact constellation.

4. AN ILLUSTRATIVE EXAMPLE

```
<xsd:elementname="flight_order">
<xsd:complexType>
<xsd:sequence>
  <xsd:elementname="odr_person" type="xs:string">
  <xsd:elementname="flight_to" type="flighttoType">
  <xsd:sequence>
  <xsd:elementname="name" type="xsd:string" use="required"/>
  <xsd:elementname="addr" type="xsd:string" use="required"/>
  </xsd:sequence>
  <xsd:elementname="flight_from" type="flightfromType">
  <xsd:sequence>
  <xsd:elementname="name" type="xsd:string" use="required"/>
  <xsd:elementname="addr" type="xsd:string" use="required"/>
  </xsd:sequence>
  <xsd:elementname="item" type="itemType">
  <xsd:sequence>
  <xsd:elementname="title" type="xsd:string" use="required"/>
  <xsd:elementname="name" type="xsd:string" use="required"/>
  <xsd:elementname="supplier" type="supplierType" use="required"/>
</xsd:sequence>
   <xsd:complexTypename="SupplierType">
   <xsd:sequence>
   <xsd:elementname="name" type="xsd:string" use="required"/>
   <xsd:elementname="supplier_id" type="xsd:string" use="required"/>
  </xsd:sequence>
<xsd:attributename="odr_id" type="xsd:string" use="required"/>
</xsd:sequence>
</xsd:complextype>
</xsd:element>
```

Figure 5. Example of XML Schema

This section illustrates a case study on a real life example. A flight management system is shown which handles the supply of items between two cities. Items are supplied by the suppliers. The XML schema of Figure 5 is converted to schema graph as shown in Figure 6. From this schema graph fact and dimension tables are generated as shown in Figure 7. Next, the proper data warehouse schema is identified. The entire process is described below.

Figure 5 is an XML schema of the given problem definition. At first the proposed methodology described in section 3.2 is applied on it to generate the schema graph. It is found that the element Flight have no predecessor. Thus it would be identified as Holder Element (HE). For the HE Flight, three Contained Elements (CE) are identified namely Item, Flight_to and Flight_from. Among these three CE's only Item has a Secondary Element (SE) namely Supplier. The attributes of every entity found from the XML schema is shown in the figure, connected with the corresponding entity. The resulting schema graph is shown in Figure 6.

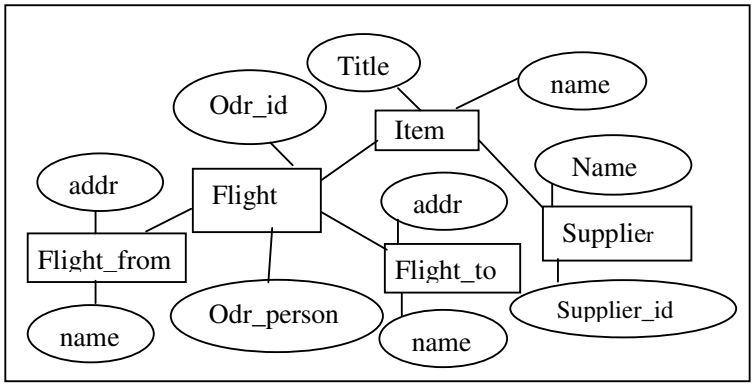
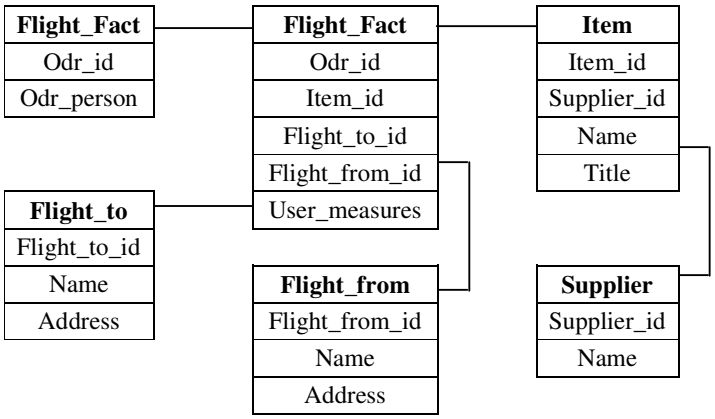


Figure 6. Schema Graph of the XML Schema

Now fact and dimension tables are identified using the procedure described in section 3.3. As the schema graph contains one HE, one fact table would be constructed namely Flight_Fact. The dimension tables for Flight_fact would be Item, Flight_to and Flight_from. The dimension table Item contains an SE namely Supplier, thus it would be connected with Item. Now primary keys are required to be identified. We assume that the entity Flight, and Supplier have the primary key hence we do not need to add any primary key (described in section 3.3). The entities Item, Flight_to and Flight_from do not have any primary key, thus the keys Item_id, Flight_to_id and Flight_from_id are added to these entities. These are shown in Figure 7.



*User_measures jn fact tables are supplied by users as their measuring unit

Figure 7. Snowflake Schema from the Schema Graph of Figure 6

The next step is to identify the proper data warehouse schema. As there is only one HE namely Flight and the CE Item has an SE namely Supplier procedure Snowflake (described in section 3.4.2) is applied to build the data warehouse schema. In the fact table there is an attribute user_measures which would be given by the users. This is also shown in Figure 7.

5. CONCLUSIONS

This paper provides a framework to convert an XML schema to ROLAP data warehouse schema. The target schema includes star schema, snowflake schema and also the fact constellation. Moreover the proposed methodology is capable to identify multiple schemas from a XML schema

if they are related. This is an interesting and emerging area of research as a number of business organizations use data warehouse for their business analysis and use XML to handle semi-structured data and also to take the advantage of using web environment. This work is a contribution to the ETL phase of data warehouse toolkit where the online transactional data is XML. Processing of data for analytical purpose is a continuous research interest. The challenge is more when the data is not structured such as in XML. The proposed methodology could be applied further on different semi-structured data to convert them suitable for data warehouse processing. Moreover if other semi-structured data are converted to XML using an intermediate step, then this methodology could be applied to transform semi-structured data to XML. Another important aspect is that if the source XML changes, then the schema structure needs to be reconstructed. Here instead of running the algorithm on the entire XML schema it could run partially only on the changed portion. This could be thought of as the maintenance phase of the proposed methodology. Thus the organizations looking to incorporate business intelligence can use this type of framework to build data warehouse architecture from heterogeneous data sources. Conversion of different types of semi-structured data to XML could be another interesting extension of the work reported here.

REFERENCES

[1] Soumya Sen, Ranak Ghosh, Debanjali Paul, Nabendu Chaki; "Integrating Related XML Data into Multiple Data Warehouse Schemas"; Proc. of the First International Conference on Information Technology Convergence and Services (ITCS 2012), Bangalore, India.

[2] Sarbani Dasgupta, Soumya Sen, Nabendu Chaki; "A Framework To Convert XML Schema to ROLAP"; Proc. of 2nd Intl. Conf. on Emerging Applications of Information Technology, 2011.

[3] Parimala N and Payel pahwa; "From XML schema to cube" International Journal of Computer Theory and Engineering; Vol. 1, No 3 August 2009.

[4] Payel pahwa and Parimala N; "Conceptual design of data warehouses from xml schemas" 2nd International Conference on Intellectual Capital, knowledge management & Organizational Learning 21-22 Nov, 2005 American University of Dubai, United Arab Emirates

[5] M. Jensen, T. Møller, and T.B. Pedersen, .Specifying OLAP Cubes On XML Data., Journal of Intelligent Information Systems, 2001.

[6] Frank S.C. Tseng, Chia Wei Chen: Integrating heterogeneous data warehouses using XML technologies, Journal of Information Science Volume-31, Issue:3 (June 2005) Page-209-229

[7] Boris Vrdoljak, Marko Banek, and Stefano Rizzi: Designing Web Warehouses from XMLSchemasY. Kambayashi, M. Mohania, W. Wöß (Eds.): LNCS 2737, pp. 89-98, 2003.

[8] Wolfgang Hummer, Andreas Bauer, Gunnar Harde: XCube – XML for Data Warehouses, DOLAP'03, November 7, 2003, USA.

[9] M. Golfarelli, S. Rizzi, and B. Vrdoljak, .Data warehouse design from XML sources., Proc. DOLAP'01, Atlanta, pp. 40-47, 2001.

[10] Data Mining Concepts and Technique,2nd Edition, Jiawei Han and Micheline Kamber, Morgan Kaufmann Publisher

[11] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau; "Extensible Markup Language (XML) 1.0 (Fifth Edition)"; W3C Recommendation; www.w3.org/TR/REC-xml

[12] Yuan Sun; Hexin Chen; Mianshu Chen; Xinying Wang; Aijun Sang; "Multi-dimension Multimedia Retrieval Model Implementation Based on XML Database" International Conference on Signal Processing Syatems, 2009

[13] Rajugan, R.; Chang, E.; Dillon, T.S.; "Conceptual Design of an XML FACT Repository for Dispersed XML Document Warehouses and XML Marts", 5th International Conference on Computer and Information Technology, 2005

[14] Ramanath, M.; Kumar, K.S.; "A rank-rewrite framework for summarizing XML documents" 24th International Conference on Data Engineering Workshop, ICDEW 2008

[15] Belen Vela; Carlos Blanco; Eduardo Fernandez; E.Marcos "Model Driven Development of Secure XML Data Warehouses: A Case Study". EDBT 2010, Lausanne, Switzerland.

[16] Daneva, M., Wieringa R "Requirements engineering for cross-organizational ERP implementation undocumented assumptions and potential mismatches", 13th IEEE International Conference on Requirements Engineering, 2005.

[17] The Data Warehouse Toolkit: The Complete Guide to Dimensional Modelling 2nd Edition, Ralph Kimball and Margy Ross, John Willy & Sons.

[18] The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data 1st Edition, Ralph Kimball and Joe Caserta, John Willy & Sons.

[19] Li Jian; Xu Bihua; "ETL Tool Research and Implementation Based on Drilling Data Warehouse" 7th Int'l Conference on Fuzzy Systems and Knowledge Discovery, Chnegdu, China

Authors

Soumya Sen is a faculty member in A. K. Choudhury School of Information Technology under University of Calcutta since March, 2009. He obtained his M. Tech. Degree in Computer science & Engineering from the University of Calcutta in 2007. His area of research is Data Warehouse and OLAP Tool. He is currently pursuing his PhD work as a part-time scholar.

Ranak Ghosh completed his M.Sc. degree in Computer Science in 2011 from Barrackpore Rastraguru Surendranath College under West Bengal State University. This work is part of his Masters dissertation work under the guidance of Soumya Sen

Debanjali Paul completed her M.Sc. degree in Computer Science in 2011 from Barrackpore Rastraguru Surendranath College under West Bengal State University. This work is part of her Masters dissertation work under the guidance of Soumya Sen

Nabendu Chaki is an Associate Professor in the Department Computer Science & Engineering, University of Calcutta, India. Besides editing several volumes in Springer proceedings, Nabendu has authored 2 text books and close to 100 refereed research papers in Journals and International conferences. His areas of research interests include distributed systems and software engineering.