

©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Conceptual Design of an XML FACT Repository for Dispersed XML Document Warehouses & XML Marts

Rajugan R.¹, Elizabeth Chang² and Tharam S. Dillon¹

¹ eXel Lab, Faculty of IT, University of Technology, Sydney, Australia

E-mail: {rajugan, tharam}@it.uts.edu.au

² School of Information Systems, Curtin University of Technology, Australia

E-mail: Elizabeth.Chang@cbs.curtin.edu.au

Abstract

Since the introduction of eXtensible Markup Language (XML), XML repositories have gained a foothold in many global (and government) organizations, where, e-Commerce and e-Business models have matured in handling daily transactional data among heterogeneous information systems in multi-data formats. Due to this, the amount of data available for enterprise decision-making process is increasing exponentially and are being stored and/or communicated in XML. This presents an interesting challenge to investigate models, frameworks and techniques for organizing and analyzing such voluminous, yet distributed XML documents for business intelligence in the form of XML warehouse repositories and XML marts. In this paper, we address such an issue, where we propose a view-driven approach for modeling and designing of a Global XML FACT (GxFACT) repository under the MDA initiatives. Here we propose the GxFACT using logically grouped, geographically dispersed, XML document warehouses and Document Marts in a global enterprise setting. To deal with organizations' evolving decision-making needs, we also provide three design strategies for building and managing of such GxFACT in the context of modeling of further hierarchical dimensions and/or global document warehouses.

1. Introduction

EXtensible Markup Language (XML) [1] has become the *defacto* standard for storing and manipulating self-describing information since its introduction in 1996. XML creates vocabularies in assisting information exchange between heterogenous enterprise data sources over the web [2, 3]. Conversely, Enterprise Content Management (ECM) is

the integration and utilization of one or more technologies, tools, and methods to capture, manage, store and deliver content across an enterprise [4], where XML is gaining momentum as the data representation and integration language.

One of the data intensive issues in ECM is the data warehousing concept that has gained in importance in recent years [5]. At the most basic level, data warehousing has been an approach adopted for management of large volumes of historical data for detailed analysis to provide crucial business intelligent (BI) for organisations in; (1) Decision Support Systems (DSS) [5, 6], Management Information Systems (MIS) [5] and (3) Executive Information Systems [6]. A data warehouse integrates large amounts of enterprise data from multiple and independent data sources consisting of operational databases into a common repository [7] for querying and analysis (using BI tools). In addition, data warehouses are designed for online analytical processing (OLAP) [5, 7-9], where the queries aggregate large volumes of data in order to detect trends and anomalies. To reduce the cost of executing aggregate queries in such an environment, warehousing systems usually pre-compute frequently used aggregates and store each materialized aggregate view [7, 10, 11] in a multidimensional data cube [7, 9, 11, 12]. These data cubes group the base data along various dimensions, corresponding to different sets of operational attributes, and compute different aggregate functions (e.g. sum, avg, min, max) on measures.

Since the introduction of OMG's Model-Driven Architecture (MDA) initiative [13], platform independent models play a vital role in system development and data engineering. Under the MDA initiative, first the model of a system is specified via an abstract notation independent of the technical or deployment specifications (i.e. Platform Independent Model or PIM) and then the PIM is mapped or

transformed into a deployment model (i.e. Platform Specific Model or PSM) by adding platform or deployment specific information into the PIM. To support MDA initiatives in ECM (i.e. data engineering, data semantics, constraints etc.), model requirements has to be specified precisely at a higher level of abstraction. This presents an opportunity to investigate data views [14] as a means of providing data abstraction and semantics in PIMs for data intensive MDA solutions.

In the context of MDA solutions for XML domains, it is still a challenging task to produce PIMs despite the flexibility and the semantic richness of the semi-structured data/schema languages (e.g. XML). This is mainly due to OO modeling languages such as OMG UML™ [15], Extended-ER [5] etc. provide insufficient modeling constructs for utilizing XML schema like data descriptions and constraints, while XML Schema lacks the ability to provide higher levels of abstraction (such as conceptual models, visual constraints) that are easily understood by humans. We note that models are often abstract representations that only keep so much of the detail as is relevant to the particular problem being considered [16]. In this context, XML Schema generally is too low a representation to permit users to interact, visualize or understand it. To rectify this situation, many researchers in work such as [17-22], have applied intuitive techniques, notations and transformation methodologies to capture XML semantics at the conceptual level. This presents an interesting challenge to look at dimensional data engineering, such as data warehouse models, under the MDA initiatives.

In this paper, we propose an view-driven *architectural construct* for conceptually modeling and designing a Global XML FACT repository (\mathcal{G} xFACT) based on logically grouped, geographically dispersed, XML FACT repositories (xFACT) [23, 24] of XML document warehouses (XDW) and XML Document Marts. Here, the proposed \mathcal{G} xFACT model is equivalent of a PIM in MDA. The initial conceptual design of xFACT (in XDW) was proposed by authors of the work in [23, 24] and Web Document Warehouse (WDW) [25], where they use Object-Oriented Conceptual Modeling (OOCM) to build a conceptual framework for XML document warehouse using a *view formalism* with *conceptual* and *logical* extensions (for XML) [14, 26].

The rest of this paper is organized as follows. In section 2, we review early work in data warehouse model and view related domains, followed by a brief introduction of our layered view model for XML in section 3. Section 4 describes an illustrative industrial case study used in this paper. In section 5, we present

our proposed global XML repository model (PIM) with its properties, followed by design methodology in section 6 and implementation options (PSM) in section 7. We conclude the paper and outline future work in section 8.

2. Related Work

Data warehousing concept that has gained in importance in recent years [7, 27, 28] as part of ECM initiatives. A data warehouse integrates large amounts of enterprise data from multiple and independent data sources consisting of operational databases into a common repository [7] for querying and analysis (using BI tools). In addition, data warehouses are designed for online analytical processing (OLAP) [7, 8, 27, 29], where the queries aggregate large volumes of data in order to detect trends and anomalies. To reduce the cost of executing aggregate queries in such an environment, warehousing systems usually pre-compute frequently used aggregates and store each materialized aggregate view [7, 10, 11] in a multidimensional data cube [7, 9, 11, 12]. These data cubes group the base data along various dimensions, corresponding to different sets of operational attributes, and compute different aggregate functions (e.g. sum, avg, min, max) on measures. To address such requirements many models have been proposed for designing data warehouses.

Since the introduction of dimensional modeling, several design techniques have been proposed to capture multidimensional data (MD) at the conceptual level. Conceptual data models for such MD, which revolves around FACTs, dimensions and hierarchies, have been extensively discussed in research and industrial literature [7, 9]. These discussions normally includes support for data warehouses and OLAP data (ROLAP, MOLAP), where MD is the feasible data model for such applications.

The early work on MD and data warehousing concepts date back to works done by W.H. Inmon et al. [6, 30-32]. Later work by Ralph Kimball's popular Star Schema [5, 8] provided the base for other well-known conceptual models such as Snowflake and StarFlake to be derived. More recent comprehensive data warehouse design models are built using Object-Oriented concepts on the foundations of the Star Schema. In [9, 33-37], and [38], two different OO modelling approaches are demonstrated where a data cube is transformed into an OO model integrating class hierarchies. The Object-Relational Star schema (O-R Star) model [39] aims to utilise Object-Relational (O-R) data model and its features for warehouse MD data and hierarchies.

For XML data, one of the early XML data warehouse implementations for web data includes the Xyleme Project [2]. The Xyleme project [40] was successful and it was made into a commercial product in 2002. It has well defined implementation architecture and proven techniques (such as materialised views) to collect and archive web XML documents into an XML warehouse for further analysis. Another approach by Fankhauser et al. [41] explores some of the changes and challenges of a document centric XML warehouse. Other works that use XML in data warehouse context includes [42, 43].

In another related work, authors focused on building a requirement driven, meaningful FACT repository in , the work on XML-view driven XML document warehouses [23, 24] and Web document warehouses [25], authors argue that, coupling these approaches with a well defined requirement-oriented [44] conceptual design methodology will help future error-free, maintainable design of such XML warehouse for large-scale XML systems.

In DW domain, views are mainly used to provide aggregate data and queries, performance (as materialized views), meta-data and OLAP queries [6, 8, 9, 12, 27, 28, 31, 33, 45-47]. Only few work has been done in the direction of using views for providing DW architectural constructs and frameworks [10, 11, 23, 24].

3. Preliminaries: The Layered View Model for XML

The 3-layered view model used to construct the GxFACT model was proposed by the authors in [14, 26]. The XML-view model has three-layers of abstraction, namely; (1) conceptual, (2) logical or schematic and (3) document or instance level. The view model is based on the postulates 1 and 2, about the real world.

Postulate 1: The term *context* refers to the domain that interests an organization as a whole. It is more than a measure and implies a meaningful collection of objects, relationships among these objects, as well as some constraints associated with the objects and their relationships, which are relevant to its applications.

Postulate 2: The term *view* refers to a certain perspective of the context that makes sense to one or more stakeholders of the organization or an organization unit at a given point in time.

The conceptual layer describes the structure and semantics of XML views in a way which is more comprehensible to human users. It hides the details of view implementation and concentrates on describing objects, relationships among the objects, as well as the

associated constraints upon the objects and relationships. Due to its abstract nature, conceptual views can be defined using any high level modeling languages such as Dillon & Tan notation [16], UML [48], XSemantic Nets [49] or Extended Entity-Relationship Model (E-ER) [27].

Definition 1: A **conceptual view** V^c is a 4-ary tuple $V^c = (V_{name}^c, V_{obj}^c, V_{rel}^c, V_{constraint}^c)$, where V_{name}^c is the name of the conceptual view V^c , V_{obj}^c is a set of objects in V^c , V_{rel}^c is a set of object relationships in V^c , and $V_{constraint}^c$ is a set of constraints associated with V_{obj}^c and V_{rel}^c in V^c .

Definition 2: Let $C = (C_{name}, C_{obj}, C_{rel}, C_{constraint})$ denote a context which consists of a context name C_{name} , a set of objects C_{obj} , a set of object relationships C_{rel} , and a set of constraints associated with its objects and relationships $C_{constraint}$. Let $\tilde{\lambda}$ be a set of conceptual operators. $V^c = (V_{name}^c, V_{obj}^c, V_{rel}^c, V_{constraint}^c)$ is called a *valid conceptual view of the context C*, if and only if the following conditions satisfy;

- (1) For any object $\forall o \in V_{obj}^c$, there exist objects $\exists o_1, \dots, o_n \in C_{obj}$, such that $o = \lambda_1 \dots \lambda_m (o_1, \dots, o_n)$ where $\lambda_1 \dots \lambda_m \in \tilde{\lambda}$. That is, o is a newly derived object from existing objects o_1, \dots, o_n in the context via a series of conceptual operators $\lambda_1, \dots, \lambda_m$ like select, join, etc.
- (2) For any constraint $\forall c \in V_{constraint}^c$, there exists a constraint $\exists c' \in C_{constraint}$ or a new constraint c'' constraints associated with V_{obj}^c or V_{rel}^c .
- (3) For any hierarchical relationship $\forall r^h \in V_{rel}^c$, there *does not exist* a relationship between one or more and V_{obj}^c and C_{obj} .
- (4) For any association relationship/dependency relationships $\forall r^a \in V_{rel}^c$, there *may exist a relationship between one or more* V_{obj}^c and C_{obj} .

The middle schema (or logical) layer describes the schema of XML views for the view implementation, using the XML Schema language. Views at the conceptual level are mapped into the views at the schema level via the transformation mechanism developed in work [17-19]. The output of this level will be in either textual (such as XML Schema language) or some visual notations that comply from the schema language (such as graph).

The document (or instance) level implies a fragment of instantiated XML data, which conforms to the corresponding view schema defined at the upper level. A detailed discussion this layered view model can be found in [14, 50].

4. An Illustrative Case Study

To illustrate our concepts, we use an example case study of a fictitious global logistics company called LWC & e-Solutions Inc. (e-Sol) that provides global logistics, warehouse space and cold storage facilities to their global (and regional) customers and collaborative partners. The e-Sol solution includes a standalone and distributed Warehouse Management System (WMS/e-WMS), and a Logistics Management System (LMS/e-LMS) on an integrated e-Business framework called e-Hub [51] for all inter-connected services for customers, business customers, collaborative partner companies, and LWC staff (for e-commerce B2B and B2C). Some real-world applications of such company, its operations and IT infrastructure can be found in [51-53].

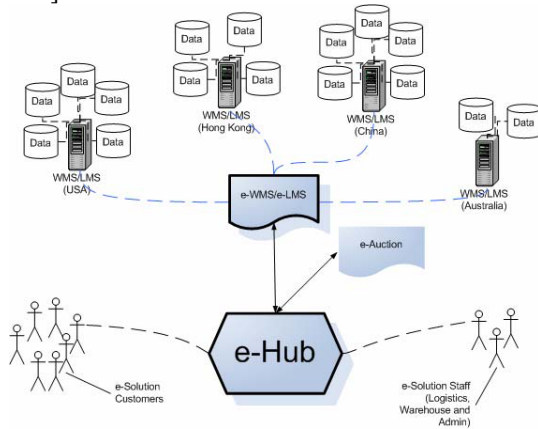


Figure 1: e-Sol context diagram

For e-Sol to support DSS, EIS and MIS, it is essential to provide a data model to support dimensional data in the context of data warehouse. Due to e-Sol's dynamic and heterogeneous nature (both system and data), the data warehouse model should support rapidly evolving new data formats (from relational, XML to propriety data scripts), at a higher-level of abstraction. For a local stakeholder/partners' perspective, the XDW model solve some of the problems faced by e-Sol. But in a global perspective, where multiple stakeholders/partner system are involved and need to support e-Sol global information demand, the role and scope of XDW has to be challenged and a new global warehouse model is inevitable and an unfortunate reality; thus the $\mathcal{G}xFACT$ model.

5. PIM: The $\mathcal{G}xFACT$ Model

In this section, we present the $\mathcal{G}xFACT$ model, its properties and implementation options.

$\mathcal{G}xFACT$ is an *architectural construct* (than an implementation or storage structure) to build and integrate multiple XML FACT repositories and XML data marts into one global XML FACT repository that provide perspectives to an organization at the global level (Fig. 2). It is an aggregated xFACT (i.e. xFACT of one or more xFACTs) to support DSS, MIS and EIS solutions in global business environments, in the context of a global XML document warehouse.

The $\mathcal{G}xFACT$ model utilizes the layered view model to provide three levels of abstraction, namely conceptual, logical/schema and document/instant levels, which in turn uses OOCM approach to data warehousing and the industry standard UML as the modeling language. The design methodology starts from initial OO operational data level to the global repository level. Since it is based on high-level modeling (conceptual and logical) semantics, it is independent of the implementation platform/storage architecture or the operational data model. Also, it does not matter which implementation model (or platform) is chosen for the $\mathcal{G}xFACT$, as far as the storage model supports native XML data. Thus we say that the model and design of $\mathcal{G}xFACT$ is platform independent (or produce Platform-Independent-Model (PIM)).

Similar to conceptual views, at the conceptual level, we can define a $\mathcal{G}xFACT$ as;

Definition 3: A global XML FACT repository ($\mathcal{G}xFACT$) GxF^c is a 4-ary tuple;

$$GxF^c = (GxF_{name}^c, GxF_{obj}^c, GxF_{rel}^c, GxF_{constraint}^c)$$

where, GxF_{name}^c is the name of the $\mathcal{G}xFACT$ repository GxF^c , GxF_{obj}^c is a set of objects in GxF^c , GxF_{rel}^c is a set of object relationships in GxF^c , and $GxF_{constraint}^c$ is a set of constraints associated with GxF_{obj}^c and GxF_{rel}^c in GxF^c .

Since a set of conceptual views (say, the i^{th} conceptual view constructed from the i^{th} XML FACT repository is $V_{XDW_i}^c$) construct the $\mathcal{G}xFACT$ from the underlying XML FACT repositories [54], GxF^c can be stated as;

$$GxF^c = \bigcup_{i=1}^n V_{XDW_i}^c$$

where, n is the finite number of XML FACT repositories from which the GxF^c is constructed.

Example 1: In e-Sol example, a conceptual views "Warehouse-Q1_revenue_China", "Warehouse-Q1_revenue_LA", "Warehouse-Q1_revenue_HK" etc.

constructed in the given context of “Warehouse_revenue”. The valid collection set is given by the e-Sol “xFACT_WMS” objects.

Example 2: Similarly, it can be shown the same for the conceptual view “Staff-salary_by_region” in the given context of “Staff_salary_pkg”.

Example 3: A conceptual view “Warehouse-Capacity-LA”, “Warehouse-Capacity-China”, “Warehouse-Capacity-Australia” and “Warehouse-Capacity-HK”, in the given context of “Warehouse-Capacity_by_season”.

Since the basic constructs are views, building hierarchical dimensions (VDim) and virtual view support for OLAP queries are built-in to the model, which can be designed using top-down approaches, based on user requirements and/or performance issues.

The GxFACT repository will; (1) provide an integrated data source for BI tools for a given context (e.g. global/regional earnings), (2) provide data and data semantics to built further (global) dimensions and dimensional hierarchies, (3) provide seamless integration of existing data warehouse sources with preserved data semantics, (4) preserve conceptual semantics of initial data warehouse environments and (5) support and reflect changing warehouse requirements at a higher level of abstraction.

Example 4: In our e-Sol example, for a GxFACT “Warehouse-Capacity”, many VDim’s such as “Regional-Warehouse-Capacity-by-Season”, “Warehouse-Capacity-by-Country”, “Profit-by-Region” etc. can be defined, providing regional and/or global perspectives.

6. The GxFACT Design Steps

Here we briefly show the steps involved in building a GxFACT model.

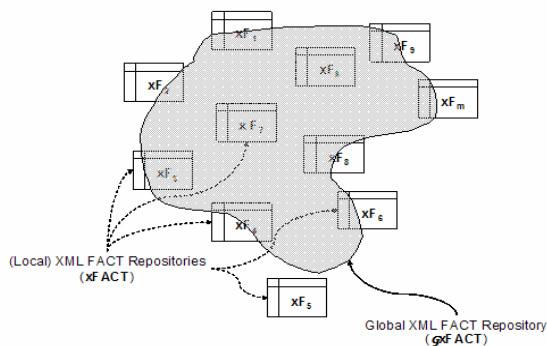


Figure 2: GxFACT domain (context diagram)

Step 1: Individual XDW models (xFACTs and associated VDims) conceptually designed and implemented for each subject-group and/or

organization unit. Priorities are given to capture local business needs and requirements (such as a large warehouse/logistics provider or departments) to support their business at the local (not global) business entity level. The xFACT and the VDims designed reflect individual and local business needs. Since the VDim constructed using views, they can reflect fast changing local business paradigm. This step is comparable to traditional data warehouse engineering, except the requirement engineering methodology is based on [44], data is XML and the warehouse architecture is based on XDW [23].

Step 2: Global XDW requirement is developed (using the requirement engineering technique in [44]) based on all existing XDW (mainly xFACT conceptual models) and the available data. Here, the main goal is to treat all participating business stakeholders (warehouse service providers, logistics providers, local business and collaborative partners, third-party logistics providers etc.) as one business entity. The output of this step is to amalgamate global business requirements into a requirement model for developing a global XDW.

Step 3: Correlated XDW (mainly xFACT) models collected and crossed-checked for duplicity and ambiguity in the context of the global XDW model.

Step 4: Multiple contexts (Fig. 2) are developed over existing (or new) xFACT repositories of the local business and enterprise stakeholders.

Step 5: For each correlated XDW, conceptual views are defined over their defined context (xFACT repository models) to refine elements of interest for the global XDW.

Step 6: Based on the requirement model (defined in Step 2) and contexts (and conceptual views) defined in Sept 3-5 above, GxFACT conceptual model is developed. Fig. 2 shows the domain of these contexts (shaded region).

Step 7: Based on the conceptual model of GxFACT, (collection of conceptual views), logical model (i.e. equivalent logical views and schemas for the conceptual views) is developed for the GxFACT. Such transformations (i.e. conceptual to logical views) are discussed in related works [14, 22].

Step 8: The GxFACT model operators (conceptual operators) are transformed into query constructs.

Step 9: The GxFACT logical model is implemented. That is, all GxFACT schemas (underlying conceptual view schemas) are made persistent and online.

Step 10: Depending on the implementation solution (PSM) adopted (see section 7), the GxFACT data is instantiated. Note: this step is comparable to the relational data warehouse ELT process [30], which

itself is a detailed process and not discussed in detail here. This step includes many operations such as; (1) data cleaning, (2) data caching, (3) data merging, (4) data aggregation, (5) data mapping, and (6) data loading to name a few.

7. PSM Options: \mathcal{G} xFACT Implementation

In \mathcal{G} xFACT model; (a) the data source for the instance data comes from one or more xFACT of either XDW or XML document marts, (b) the Meta-Data (due to the nature of XML/XML Schema) is embedded within the source data and (c) the \mathcal{G} xFACT is designed using (conceptual and logical) views. Due these factors, we provide three implementation solution models to implement the Global XDW (namely the \mathcal{G} xFACT and the associated VDims).

Option 1: Fully *persistent* (or materialized) \mathcal{G} xFACT repository with dimensions. In this option, the \mathcal{G} xFACT (the collections of views that form the \mathcal{G} xFACT) are fully materialized. New VDims can be defined over this repository as in simple XDW model. Also, if needed, depending on user requirements and performance, some VDims can be materialized (e.g. to support OLAP queries).

This option is preferred in a situation, where dimensional definitions are of dynamic nature (user query, to support third-part analytical tool etc.) and high-performance computing power (and network resources) are in abundance. This also suits well where the \mathcal{G} xFACT should remain reasonably static (e.g. due to the underlying XDW data sources and connectivity) and updated constantly in regular intervals to maintain data accuracy [55].

Option 2: Non-persistence (or non-materialized) \mathcal{G} xFACT repository with persistence global dimensions. This option is unorthodox, where, \mathcal{G} xFACT logical model is implemented (i.e. schemas, environment parameters etc.), but data is not materialized at the \mathcal{G} xFACT level. This situation is comparable to a view definition stored in a relational model. But all the VDims are defined and materialized with their data. Here, \mathcal{G} xFACT serves as a meta-data repository than a XDW repository.

This option is preferred when an organization has fixed warehouse requirements (at the global level) and wide-range of high-performance storage solutions (not computational power, such as grid or cluster computing). Also, this solution is feasible, if the underlying operational data sources are updated over a longer term than in regular, short (weekly or monthly) intervals. Another advantage of such option is that, since all the dimensions are already materialized (i.e.

all complicated query processing is done and data is readily available for end-users), end-users do not require high-performance computing power, thus, suited well for regions that suffer from such issues.

Option 3: Here, it is a combination of option 1 and 2, where predefined sections of the \mathcal{G} xFACT repository (i.e. selected views) and selected VDims (i.e. dimensional views) are materialized based on business (and performance) requirements.

In making a decision on which option to choose, the following factors should be considered; (1) \mathcal{G} xFACT requirements, (2) availability of computing power (and associated resources), (3) end-user computing resources, (4) end-user knowledge, (5) support for in-house/third party analytical tools, (7) estimated size (and predicted growth rate) of the \mathcal{G} xFACT and (6) required performance level.

For example, if a section of the business or regional XDW (or data availability) suffers from lack of computing and network resources, \mathcal{G} xFACT/VDim sections associated with such data may be materialized to improve data availability and/or performance for the overall global XDW. Note: In deciding option 3, in addition to warehouse and business requirements (steps 1-3 in section 6), warehouse operational requirements must be considered (which section/(s) and/or VDim to materialize etc.) in designing the \mathcal{G} xFACT conceptual model.

Example 5: In our e-Sol example, in addition to XDWs for local warehouse owners, regional \mathcal{G} xFACT (e.g. Europe, USA, China) can be built to support growing business demand and/or if a requirement exists, where warehouse/logistics turn-over is very high. E.g. "US-Logistics-Orders".

8. Conclusion and Future Work

In this paper, we presented an intuitive, layered view-driven, *architectural construct* (equivalent to PIMs in MDA) to conceptually model, design and implement a global XML FACT repository, \mathcal{G} xFACT. Such repository will combine organizations' existing warehouse solutions (XML Document Warehouses and XML marts) to provide an integrated (or global) perspective for DSS, MIS and EIS.

For future work, some further issues deserve investigation. First, the investigation into OLAP support in \mathcal{G} xFACT. Second is the derivation of a formal \mathcal{G} xFACT (and associated VDim) model. Finally, is the formulation of a valid empirical study to consider model transformation and model mapping formalism between the traditional xFACT models and the \mathcal{G} xFACT model.

9. References

- [1] W3C-XML, "Extensible Markup Language (XML) 1.0, (<http://www.w3.org/XML/>)," 3 ed: The World Wide Web Consortium (W3C), 2004.
- [2] Lucie-Xyleme, "Xyleme: A Dynamic Warehouse for XML Data of the Web," Int. Database Engineering & Applications Symposium (IDEAS '01), Grenoble, France, 2001.
- [3] J. Pokorn'y, "XML Data Warehouse: Modelling and Querying," Proc. of the Baltic Conf. (BalticDB-IS '02), 2002.
- [4] "The ECM Association (<http://www.aiim.org/index.asp>)," AIIM, 2005.
- [5] R. Elmasri and S. Navathe, *Fundamentals of database systems*, 4th ed. New York: Pearson/Addison Wesley, 2004.
- [6] P. Gray and H. J. Watson, *Decision Support in The Data Warehouse*. USA: Prentice Hall PTR, 1998.
- [7] L. Feng and T. S. Dillon, "Using Fuzzy Linguistic Representations to Provide Explanatory Semantics for Data Warehouses," *IEEE Transactions on Knowledge and Data Engineering (TOKDE)*, vol. 15, No. 1, pp. 86-102, 2003.
- [8] R. Kimball and M. Ross, *The data warehouse toolkit : the complete guide to dimensional modeling*, 2nd ed. New York: Wiley, 2002.
- [9] J. Trujillo, et al., "Applying UML For Designing Multidimensional Databases And OLAP Applications. 13-36," *Advanced Topics in Database Research, Idea Group Publication*, vol. 2, pp. 13-36, 2003.
- [10] D. Theodoratos and T. Sellis, "Dynamic Data Warehouse Design," 1st Int. Conf. on DaWaK '99, Italy, 1999.
- [11] V. Gopalkrishnan, Q. Li, and K. Karlapalem, "Star/Snow-flake Schema Driven Object-Relational Data Warehouse Design and Query Processing Strategies," 1st First Int. Conf. on DaWaK '99, Florence Italy, 1999.
- [12] A. Gupta, I. S. Mumick, and (eds), *Materialized views: techniques, implementations, and applications*: MIT Press, 1999.
- [13] OMG-MDA, "The Architecture of Choice for a Changing World®, MDA Guide Version 1.0.1 (<http://www.omg.org/mda/>)," OMG, 2003.
- [14] R.Rajugan, E. Chang, T. S. Dillon, and F. Ling, "A Three-Layered XML View Model: A Practical Approach," 24th Int. Conf. on Conceptual Modeling (ER '05), Klagenfurt, Austria, 2005.
- [15] OMG-UML™, "UML 2.0 Final Adopted Specification (<http://www.uml.org/#UML2.0>)," 2003.
- [16] T. S. Dillon and P. L. Tan, *Object-Oriented Conceptual Modeling*: Prentice Hall, Australia, 1993.
- [17] L. Feng, E. Chang, and T. S. Dillon, "A Semantic Network-based Design Methodology for XML Documents," *ACM Transactions on Information Systems (TOIS)*, vol. 20, No 4, pp. 390 - 421, 2002.
- [18] L. Feng, E. Chang, and T. S. Dillon, "Schemata Transformation of Object-Oriented Conceptual Models to XML," *Int. Journal of Computer Systems Science & Engineering*, vol. 18, No. 1, pp. 45-60, 2003.
- [19] R. Conrad, D. Scheffner, and J. C. Freytag, "XML conceptual modeling using UML," 19th Int. Conf. on ER '00, USA, 2000.
- [20] R. Xiaou, T. S. Dillon, E. Chang, and L. Feng, "Modeling and Transformation of Object-Oriented Conceptual Models into XML Schema," 12th Int. Conf. on Database and Expert Systems Applications (DEXA '01) 2001, 2001.
- [21] R. Xiaou, T. S. Dillon, E. Chang, and L. Feng, "Mapping Object Relationships into XML Schema," Proc. of OOPSLA Workshop on Objects, XML and Databases, 2001.
- [22] Rajugan R., E. Chang, T. S. Dillon, and L. Feng, "XML Views, Part III: Modeling XML Conceptual Views Using UML," 7th Int. Conf. on Enterprise Information Systems (ICEIS '05), Miami, USA, 2005.
- [23] V. Nassis, R.Rajugan, T. S. Dillon, and W. Rahayu, "XML Document Warehouse Design," 6th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK '04), Zaragoza, Spain, 2004.
- [24] V. Nassis, R.Rajugan, T. S. Dillon, and W. Rahayu, "Conceptual and Systematic Design Approach for XML Document Warehouses," *Int. Journal of Data Warehousing and Mining*, vol. 1, No 3, 2005.
- [25] V. Nassis, et al., "A Systematic Design Approach for XML-View Driven Web Document Warehouses," Int. Workshop on Ubiquitous Web Systems and Intelligence (UWSI '05), Singapore, 2005.
- [26] R.Rajugan, E. Chang, T. S. Dillon, and F. Ling, "XML Views: Part 1," 14th Int. Conf. on DEXA '03, Prague, Czech Republic, 2003.
- [27] R. Elmasri and S. B. Navathe, *Fundamentals of database systems*, 3 ed: Addison-Wesley, Reading, Mass. Harlow, 2000.
- [28] P. Ponniah, *Data Warehousing Fundamentals: A Comprehensive Guide for IT professionals*. NY: John Wiley & Sons Inc., 2001.
- [29] R. Kimball and J. Caserta, *The data warehouse ETL toolkit : practical techniques for extracting, cleaning, conforming, and delivering data*. Hoboken, NJ: Wiley, 2004.
- [30] Jacek Blazewicz, et al., *Handbook on Data Management in Information Systems*: Springer, Berlin ; New York, 2003.
- [31] M. Humphris, M. W. Hawkins, and M. C. Dy, *Data Warehousing: Architecture & Implementation*. USA: Prentice Hall PTR, 1999.
- [32] W. H. Inmon, C. Imhoff, and G. Battas, *Building the operational data store*. New York, USA: John Wiley & Sons, 1996.
- [33] J. Trujillo, M. Palomar, J. Gomez, and I.-Y. Song, "Designing Data Warehouses with OO Conceptual Models," in *IEEE Computer Society, "Computer"*, 2001, pp. 66-75.
- [34] S. Lujan-Mora, J. Trujillo, and I.-Y. Song, "Extending the UML for Multidimensional Modeling," Fifth Int. Conf. on the Unified Modeling Language and its applications (UML '02), Dresden, Germany, 2002.
- [35] S. Luján-Mora, J. Trujillo, and P. Vassiliadis, "Advantages of UML for Multidimensional Modeling," Proc. of the 6th Int. Conf. on Enterprise Information Systems (ICEIS '04), Porto, Portugal, April, 2004.
- [36] S. Lujan-Mora, J. Trujillo, and I.-Y. Song, "Multidimensional Modeling with UML Package Diagrams," Proc. of the 21st Int. Conf. on Conceptual Modeling (ER '02), 2002.
- [37] S. Luján-Mora, P. Vassiliadis, and J. Trujillo, "Data Mapping Diagrams for Data Warehouse Design with UML," 23rd Int. Conf. on Conceptual Modeling (ER '04), Shanghai, China, 2004.
- [38] A. Abelló, J. Samos, and F. Salto, "Understanding facts in a multidimensional object-oriented model," 4th Int. Workshop on Data Warehousing and OLAP (DOLAP '01), 2001.
- [39] W. Rahayu, T. S. Dillon, S. Mohammed, and D. Taniar, "Object-Relational Star Schemas," 13th IASTED Int. Conf. on Parallel & Disubited Computing and Systems (PDCS '01), LA, USA, 2001.
- [40] Xyleme, "Xyleme Project (<http://www.xyleme.com/>)," 2001.
- [41] P. Fankhauser and T. Klement, "XML for Data Warehousing Changes & Challenges," DaWaK 2003, Prague, 2003.
- [42] M. Golfarelli, S. Rizzi, and B. Vrdoljak, "Data warehouse design from XML sources," Proc. of the 4th ACM Int. workshop on Data warehousing and OLAP, Atlanta, Georgia, USA, 2001.
- [43] E. Medina, S. Luján-Mora, and J. Trujillo, "Handling Conceptual Multidimensional Models Using XML through DTDs," Proc. of the 19th British National Conf. on Databases: Advances in Databases, 2002.
- [44] V. Nassis, T. S. Dillon, W. Rahayu, and R.Rajugan, "Goal-Oriented Requirement Engineering for XML Document Warehouses," in *Processing and Managing Complex Data for Decision Support*, J. Darmont and O. Boussaid, Eds.: Idea Group Publishing, 2005.
- [45] T. Debevoise, *The Data Warehouse Method: Integrated Data Warehouse Support Environments*. USA: Prentice Hall PTR, 1999.
- [46] T. Johnson, "Data warehousing," in *Handbook of massive data sets*: Kluwer Academic Publishers, 2002, pp. 661-710.
- [47] J. Trujillo and S. Luján-Mora, "A UML Based Approach for Modeling ETL Processes in Data Warehouses," Int. Conf on Conceptual Modeling (ER '03), 2003.
- [48] OMG-UML™, "Unified Modeling Language™ (UML) Version 1.5 Specification," OMG 2003.
- [49] Rajugan R., E. Chang, L. Feng, and T. S. Dillon, "Semantic Modelling of e-Solutions Using a View Formalism with Conceptual & Logical Extensions," 3rd Int. IEEE Conf. on Industrial Informatics (INDIN '05), Perth, Australia, 2005.
- [50] R.Rajugan, E. Chang, T. S. Dillon, and F. Ling, "A Layered View Model for XML Repositories & XML Data Warehouses," The 5th Int. Conf. on Computer and Information Technology (CIT '05), China, 2005.
- [51] E. Chang, et al., "A Virtual Logistics Network and an e-Hub as a Competitive Approach for Small to Medium Size Companies," 2nd Int. Human.Society@Internet Conf., Seoul, Korea, 2003.
- [52] E. Chang, et al., "Virtual Collaborative Logistics and B2B e-Commerce," e-Business Conf., Duxon Wellington, NZ, 2001.
- [53] ITEC, "iPower Logistics (<http://www.logistics.cbs.curtin.edu.au/>)," 2002.
- [54] R.Rajugan, E. Chang, and T. S. Dillon, "Conceptual Design of an XML-View Driven, Global XML Fact Repository," 1st Int. Workshop on Data Management in Global Data Repositories (GRep '05), Denmark, 2005.
- [55] M. K. Mohania, K. Karlapalem, and Y. Kambayashi, "Data Warehouse Design and Maintenance through View Normalization," 10th Int. Conf. on Database and Expert Systems Applications (DEXA '99), Florence, Italy, 1999.