MODELING DIMENSIONS IN THE XDW MODEL
A LVM-Driven Approach

R. Rajugan, Elizabeth Chang and Tharam S. Dillon
Digital Ecosystems and Business Intelligence Institute (DEBII), Curtin Univeristy of Technology, Australia

Keywords: XML, data/document warehouse, views, Object-Oriented conceptual models, Layered View Model (LVM), XML views.

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 maturated in handling daily transactional data among heterogeneous information systems. 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 analysing such voluminous, yet distributed XML documents for business intelligence in the form of XML warehouse repositories and XML marts. In our previous work, we proposed a *Layered View Model (LVM)* driven, conceptual modelling framework for the design and development of an XML Document Warehouse (XDW) model with emphasis on conceptual and logical semantics. There, we presented a view-driven framework to conceptually model and deploy meaningful XML FACT repositories in the XDW model. Here, in this paper, we look at the hierarchical dimensions and their theoretical semantics used to design, specify and define dimensions over an XML FACT repository in the XDW model. One of the unique properties of this LVM-driven approach is that the dimensions are considered as first-class citizens of the XDW conceptual model. Also, here, to illustrate our concepts, we use a real-world case study example; a logically grouped, geographically dispersed, XDW model in the context of a global logistics and cold-storage company.

1 INTRODUCTION

Data Warehousing (DW) has been an approach adopted for handling large volumes of historical data for detailed analysis and management support. Transactional data in different databases is cleaned, aligned and combined to produce good data warehouses. 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 intelligence (BI) for organisations in: (i) Decision Support Systems (DSS) (Elmasri & Navathe 2004; Gray & Watson 1998), (ii) Management Information Systems (MIS) (Elmasri & Navathe 2004) and (iii) Executive Information Systems (Gray & Watson 1998).

A data warehouse integrates large amounts of enterprise data from multiple and independent data sources consisting of operational databases into a common repository (Feng & Dillon 2003) for querying and analysis (using BI tools). In addition, data warehouses are designed for online analytical processing (OLAP) (Elmasri & Navathe 2004; Feng & Dillon 2003; Kimball & Ross 2002; Trujillo, Luján-Mora & Song 2003), 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 (Feng & Dillon 2003; Gopalkrishnan, Li & Karlapalem 1999; Theodoratos & Sellis 1999) in a multidimensional data cube (Feng & Dillon 2003; Gopalkrishnan, Li & Karlapalem 1999; Gupta, Mumick & (eds) 1999; Trujillo, Luján-Mora & Song 2003). 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.

Rajugan R., Chang E. and S. Dillon T. (2007).
MODELING DIMENSIONS IN THE XDW MODEL - A LVM-Driven Approach.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - DISI*, pages 60-70
DOI: 10.5220/0002372700600070

In traditional data warehouse terminology, the dimensional model is represented using FACTs and dimensions. A FACT is a business performance measurement usually numeric in nature and a dimension refers to an independent entity that serves as an entry point and/or mechanism to extract meaningful measurements form the associated FACT (Elmasri & Navathe 2004; Kimball & Caserta 2004; Kimball & Ross 2002). Also, depending on the dimensional model (i.e. OO (Giovinazzo 2000), Star Schema, O-R star (Mohammed 2001)) further terminologies (e.g. FACT dimensions, shared dimensions, etc.) are defined to elaborate some of additional features of those models. In the relational model, the popular dimensional model is the Kimbal et al. Star Schema model (Kimball & Ross 2002), where the FACT and the dimensions are represented using tables (and materialized views), referred to as FACT and dimension tables. A FACT table is a non-normalized table with the numeric performance measurements characterized by a group of (foreign) keys drawn from the dimensional tables that form a composite (foreign key).

Since its introduction in 1996, eXtensible Markup Language (XML) (W3C-XML 2004) has become the *defacto* standard for storing and manipulating self-describing information (meta-data), which creates vocabularies in assisting information exchange between heterogenous data sources over the web (Pokorn'y 2002). Due to this, there is considerable work to be achieved in order to allow electronic document handling, electronic storage, retrieval and exchange. It is envisaged that XML will also be used for logically encoding documents for many domains. Hence, it is likely that a large number of XML documents will populate the would-be repository and several disparate transactional databases. 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 (ECM-AIIM 2005), 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 (Elmasri & Navathe 2004).

The concern of managing large amounts of XML document data raises the need to explore the data warehouse approach through the use of XML document marts and XML document warehouses. Since the introduction of dimensional modelling which, evolves around facts and dimensions, several design techniques have been proposed to capture multidimensional data (MD) at the conceptual level. Ralph Kimball's Star Schema (Kimball & Ross 2002) proved most popular, from which well-known conceptual models SnowFlake and StarFlake were derived. More recent comprehensive data warehouse design models are built using Object-Oriented concepts (structural relationships, Object cubes or data cubes) on the foundation of Star Schema. In (Abelló, Samos & Saltor 2001; Lujan-Mora, Trujillo & Song 2002a, 2002b; Luján-Mora, Trujillo & Vassiliadis 2004; Trujillo et al. 2001) 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 (Mohammed 2001; Rahayu et al. 2001) aims to envisage data models and their object features, focusing on hierarchical dimension presentation, differentiation and their various sorts of embedded hierarchies.

These models, both object and relational, have a number of drawbacks, namely: data-oriented without sufficient emphasis or capturing user requirements; extensions of semantically poor relational models (Star, Snowflake models) (Giovinazzo 2000; Rahayu et al. 2001), original conceptual semantics are lost before building data warehouses as the operational data source is relational, further loss of semantics resulting from oversimplified dimensional modelling (Nunamaker, Chen & Purdin 1991; Rahayu et al. 2001), time consuming if additional data semantics are required to satisfy evolving user requirements, and complex query design and processing is needed, therefore maintenance is troublesome (Inmon, Imhoff & Battas 1996; Mohania, Karlapalem & Kambayashi 1999). In applying these approaches to the design of XML document warehouses, it is important to consider XML's non-scalar, set-based and semi-structured nature. Traditional design models lack the ability to utilise or represent XML design level constructs in a well-defined abstract and implementation-independent form.

Thus, to resolve some of these issues, in our work, we consider a conceptual modelling (that includes user requirement modelling (Vicky Nassis et al. 2006) approach in proposing a document warehouse model for XML. In the following sections we present this model and the associated semantics in detail.

This paper is organised as follows: Section 2 provide the motivation behind the XDW model proposal, followed by a brief description our research in Section 3. Section 4 provides XDW model semantics in detail, followed by the description of the illustrative real-world case study example used in this paper. Section 6 concludes this paper with some discussion on our future research directions.

2 MOTIVATION

Only recently, data warehouse models have focused on incorporating conceptual semantics and user requirements as part of the model specification. This kind of work is still in the early stages; only a few such works consider conceptual semantics (in contrast to operational data oriented classical data warehouse designs such as the Star Schema (Kimball & Ross 2002) model) and are focused on both conceptual and user requirements as part of the DW design process.

In this paper, we present our approach to this problem: a *view-driven* conceptual framework for developing dimensional conceptual models for XML documents.

Our work is radically different from existing works such as (Gopalkrishnan, Li & Karlapalem 1999; Jeong & Hsu 2001; Lucie-Xyleme 2001; Luján-Mora, Trujillo & Vassiliadis 2004; Luján-Mora, Vassiliadis & Trujillo 2004; Medina, Luján-Mora & Trujillo 2002; Mohammed 2001; Mohania, Karlapalem & Kambayashi 1999). This is because, in these DW models, views are mainly used to provide aggregate data and queries, performance (as materialized views), meta-data and OLAP queries (Gupta, Mumick & (eds) 1999; Humphries, Hawkins & Dy 1999; Mohania, Karlapalem & Kambayashi 1999; Trujillo, Luján-Mora & Song 2003). Little work has been done in the direction of using views for providing DW architectural constructs and frameworks (Gopalkrishnan, Li & Karlapalem 1999; Theodoratos & Sellis 1999).

For XML data, one of the early XML data warehouse implementations for web data includes the Xyleme Project (Lucie-Xyleme 2001). The Xyleme project (Xyleme 2001) was successful and it was made into a commercial product. 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. (Fankhauser & Klement 2003) explores some of the changes and challenges of a document-centric XML warehouse. Other works that use XML in a data warehouse context include (Golfarelli, Rizzi & Vrdoljak 2001; Medina, Luján-Mora & Trujillo 2002). Our research is different from these works as views are used to model and design dimensional data instead of using views for the purpose of providing data granularity, dimensional refinements and/or for performance (e.g. materialized views). The XDW is designed for XML data and documents by incorporating XML specific data semantics.

Our work is also different from approaches such as OMG's CWM approaches (OMG-CWM 2001), where metadata model specifications based on MOF (OMG-MOF™ 2003) were proposed for developing a (mostly relational) data warehouse conceptual model. Conversely, the works such (Lujan-Mora, Trujillo & Song 2002a, 2002b; Luján-Mora, Trujillo & Vassiliadis 2004; Luján-Mora, Vassiliadis & Trujillo 2004; Trujillo, Luján-Mora & Song 2003; Trujillo et al. 2001), where conceptual models of a (mostly relational) data warehouse are developed using Object-Oriented (OO) techniques and languages were proposed. Though they are similar to our work from a conceptual modelling point of view, the works do not include: (a) an architectural framework to develop a common framework for different data domains; (b) explicit data warehouse requirement specification and notational representation of such user requirements; and (c) constructs to model semi-structured (e.g. XML) data model specific semantics such as ordering. But in the work, some new directions have been proposed to support OO concepts in the traditional FACT driven data warehouse models.

As stated before, in this research, we look at utilizing ***views*** in the Layered View Model (LVM) for XML (R.Rajugan 2006; R.Rajugan et al. 2005) as the foundation for developing conceptual framework for dimensional modelling rather than representing aggregate and/or dimensional queries (and query wrappers). Also, the design of the dimensional model is focused on capturing and modelling user requirements (Vicky Nassis et al. 2006; Vicky Nassis et al. 2006), as opposed to developing dimensional models using available operational data and the associated semantics (Kimball & Caserta 2004; Luján-Mora & Trujillo 2004). In summary, the main motivation for XDW research includes:

(i) *User requirements*: separation of operational data semantics and data warehouse user requirements in the case of XML data and document,

(ii) *Top-down approach*: separation of implementation concerns (data format, structure, etc.) from (XML) data warehouse conceptual models,

(iii) *Expressiveness*: formulation of dimensional semantics that are capable of expressively modelling XML data (both data and document centric) semantics,

(iv) *XML*: providing dimensional semantics that can be expressed and described using XML (and XML Schema) itself and

(v) *Views in the LVM*: investigating application of the LVM for dimensional modelling in achieving (ii) – (iv) above.

Another motivation is the design of DW using conceptual semantics such as in OMG's Model-Driven Architecture (MDA) initiative (OMG-MDA 2003). Since the introduction of the MDA initiative, 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 have to be specified precisely at a higher level of abstraction. This presents an opportunity to investigate conceptual views as a means of providing data abstraction and semantics in PIMs for data intensive MDA solutions.

It should be noted that, though we refer to XDW as an XML Document Warehouse, the concepts presented here are common to both data-centric and document-centric XML documents.

3 OUR WORK: LVM-DRIVEN XML DOCUMENT WAREHOUSE (XDW) MODEL

The XDW model proposed in this research is composed of four design levels, namely:

(i) XDW requirements level
(ii) XDW conceptual level
(iii) XDW logical (or schema) level
(iv) XDW document (or instance) level

Here, except for the requirements level, the other three levels are analogous to the layers of abstraction in the LVM. The XDW requirements level, in addition to the layers of abstraction, enforces user requirements in the form of (XML) Warehouse Requirements (WR) and User Requirement (UR) (Vicky Nassis et al. 2006; Nassis et al. 2005a), which complements the XDW conceptual model. A context diagram of this model is given in Fig. 1. Thus, the uniqueness of the XDW model is also in its approach to capturing and specifying data warehouse requirements. This is because, traditionally, a data warehouse model is heavily constrained by the available operational data and its structure, as warehouse modellers and designers design a data warehouse using bottom-up approaches (or reverse engineering warehouse requirements), working from operational data to the warehouse conceptual model.

Thus, it should be noted here that, in comparison with traditional data warehouse requirements, as unique to XDW design, we first develop the UR model using specialized notations that are independent of the operational data and/or data structures. Also, the UR model is developed first before constructing the conceptual model of the XDW, but it is *iteratively* validated against the available operational data, (conceptual) model and/or structures. By adopting this approach, we intend to model and represent user requirements that are *valid* yet independent of the operational data.

Also, in addition to adopting user requirement driven XDW design, the XDW model outlined below, to best of our knowledge, is unique in its kind as it is utilizes XML itself (together with XML Schema) to provide; (i) structural constructs, (ii) metadata, (iii) validity, and (iv) expressiveness (via refined granularity and class decompositions).

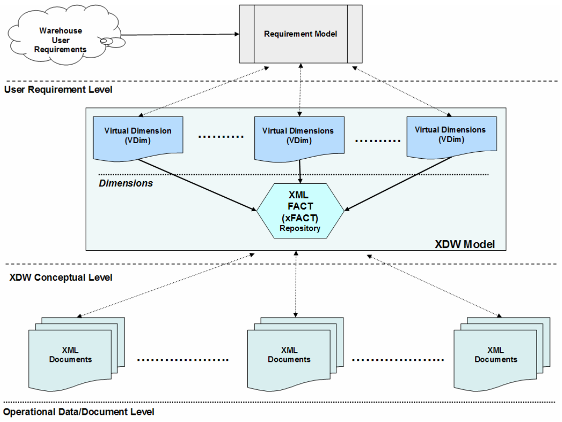

Figure 1: The XDW Model (context diagram).

As shown in Fig. 1, the first design level is the user requirement level which has two components, namely: (a) warehouse requirements, and (b) user requirements.

The second design level (Fig. 1) is the XDW conceptual model that has two main components, namely: (a) the XML FACT Repository (*x*FACT) and (b) the Virtual Dimensions (VDim).

The third level is the logical model of the XDW, where the schemata transformation of the *x*FACT and the associated VDims to (XML) schemas occurs. The fourth level is the transformation of VDim construct (i.e. conceptual operators (R.Rajugan 2006)) to document level query expressions using one or more native or embedded query languages (e.g. XQuery, SQL '03).

In this paper, we focus only on the conceptual and the theoretical semantics of the VDims. xFACT model is discussed in detail in (R.Rajugan, Chang & Dillon 2005) and the modelling and transformation of VDim from conceptual level to logical (and instance level) transformation are analogous to the views in the LVM, as discussed in detail in (R.Rajugan et al. 2005, 2006). Thus, we do not include these discussions in this paper.

3.1 XDW Conceptual Level

As stated earlier, the XDW conceptual model is composed of: (a) an XML FACT repository (*x*FACT); and (b) a collection of associated, logically grouped ***conceptual views*** that satisfies one or more user requirements given in the UR model. The *x*FACT is a snapshot of the underlying transactional system(s) for a given *context*.

As defined earlier in (R.Rajugan, Chang & Dillon 2005; R.Rajugan et al. 2005), a *context* is more than a measure or an item that is of interest for the organization as a whole. In classical data warehouse models, a context is normally modelled as an ID packed FACT and associated data perspectives as dimensions. Usually, due to constraints of the relational model, a FACT will be collapsed to a single table, with IDs of its dimension(s), thus emulating (with combination of one or more dimension(s)) a data cube (or dimensional data). A complex set of queries are needed to extract information from the FACT-Dimension model. But, in regards to XML, a context is more than a flattened FACT (or simply referred to as *meaningless* FACT) with embedded semantics and constraints. It will also have embedded relationships such as those featured in OO models and semi-structured data such ordered composition, exclusive disjunction etc. in addition to non-relational constructs such as set, list, and bag. Therefore, we argue that, a FACT structure similar to a (relational) FACT table without the required semantics does not provide semantic constructs that are needed to accommodate an XML *context*.

The role of conceptual views is to provide perspectives to the document hierarchy stored in the *x*FACT repository. Since conceptual views can be grouped into logical groups, each group is very similar to that of a ***subject area*** (or class categories) (Dillon & Tan 1993) in OO conceptual modelling techniques. Each subject-area in the XDW model is referred to as a cluster of *Virtual Dimensions* (VDim) in accordance with dimensional models. VDim is called *virtual*; that is, since it is modelled using XML *conceptual views* (which are *imaginary* XML documents) in the LVM and behaves as a dimension for the given *x*FACT. In this paper we only elaborate on VDims. A detailed discussion on xFACT can be found in our work in (R.Rajugan, Chang & Dillon 2005).

3.2 View-Driven Virtual Dimensions

A user requirement, which is captured and specified in the XDW requirement model (namely UR model), is transformed into one or more conceptual views in the LVM, which are referred to as *Virtual Dimension/s*, (VDim) in association with the *x*FACT. These are typically *views* involving aggregation or *perspectives* of the underlying *x*FACT, which serves as the pre-defined context.

A valid user requirement is such that, it can be satisfied by one or more *conceptual views* for a given context (i.e. *x*FACT). But in the case where for a given user requirement there is no transactional document or data fragment to satisfy it, further enhancements are necessary to make the requirement feasible to model with a certain *x*FACT. Therefore, modelling and specifying VDim is an iterative process, where user requirements are validated against the *x*FACT in conjunction with the operational data and data semantics. Thus, VDim is an additional elaboration, extraction and/or specification of the required information from the *x*FACT (thus, from the aggregated operational data). It should be noted here that, conceptual views are *first-class* citizens of the conceptual model. Therefore, since a VDim is a conceptual view, VDim is also a *first-class* citizen of the conceptual model.

VDim can be materialized (for data refinement or for the purpose of performance issues such as relational views in classical Star model) or aggregated further by defining additional conceptual views to refine and/or satisfy further user requirements.

4 XDW MODEL SEMANTICS

In this section, we present some of the formal semantics associated with the XDW model (without the UR model). Since the proposed model is driven by views of LVM, some of the concepts and definitions are extensions and/or an elaboration of the concepts and definitions presented in (R.Rajugan et al. 2005).

The XDW conceptual model consists of an *x*FACT repository and multiple hierarchical dimensions. Thus, at first glance, the XDW is analogous to the Star/Snowflake schema (Gopalkrishnan, Li & Karlapalem 1999) of the relational model, or the Operational Data Store (ODS) (Inmon, Imhoff & Battas 1996) model, except that both the *x*FACT and the VDims are modelled using views in the LVM. Also, since *x*FACT is more complex than a relational FACT table, it can be considered as one context (e.g. sales) that is of interest to the organization, with multiple sub-contexts (such as regional-sales, sales-by-city, sales-by-store etc.). Therefore it can be shown that there exists a many-to-many (m:n) relationship between one *x*FACT and a VDim (or a VDim hierarchy).

4.1 xFACT and View-Driven VDim

Let XML FACT repository (*x*FACT) be denoted as x_{FACT} and a virtual dimension (VDim) denoted as V^{Dim} in the XDW model. In work with LVM for XML, we defined a *conceptual view* using a *context*. Since each VDim is a conceptual view, by definition, a virtual dimension V^{Dim} can be defined as:

Definition 1. A virtual dimension is a conceptual view V^{Dim}, such that V^{Dim} is a 4-ary tuple of $V^{Dim}_{name}, V^{Dim}_{obj}, V^{Dim}_{rel}$ and $V^{Dim}_{constraint}$, where V^{Dim}_{name} is the name of the virtual dimension V^{Dim}, V^{Dim}_{obj} is a set of objects in V^{Dim}, V^{Dim}_{rel} is a set of object relationships in V^{Dim}, and $V^{Dim}_{constraint}$ is a set of constraints associated with V^{Dim}_{obj} and V^{Dim}_{rel} in V^{Dim}.

$$V^{Dim} = (V^{Dim}_{name}, V^{Dim}_{obj}, V^{Dim}_{rel}, V^{Dim}_{constraint}) \quad (1)$$

If one considers an xFACT to be a context, it can be shown as in definition 2, below using the definition of context presented in (R.Rajugan et al. 2005, 2006) as:

Definition 2. An XML FACT (*x*FACT) repository xF is defined such that, xF consists of a *x*FACT name xF_{name}, a set of objects xF_{obj} ,, a set of object relationships xF_{rel}, and a set of constraints associated with its objects and relationships $xF_{constraint}$.

$$xF = (xF_{name}, xF_{obj}, xF_{rel}, xF_{constraint}) \quad (2)$$

Similar to the definition of a valid conceptual view (R.Rajugan et al. 2005, 2006), here we can define a valid virtual dimension as:

Definition 3. A virtual dimension V^{Dim} called a *valid* virtual dimension for a given XML FACT (*x*FACT) repository xF, if and only if for any object $\forall obj \in V^{Dim}_{obj}$, there exist objects $\exists obj_1, obj_2, ..., obj_n \in xF_{obj}$, such that $obj = \lambda_1 .... \lambda_m (obj_1, ..., obj_n)$, where $\lambda_1 .... \lambda_m \in \lambda$ and λ be a set of conceptual operators. That is, obj is a newly derived object from existing objects $obj_1, obj_2, ..., obj_n$ in xF via a series of conceptual operators $\lambda_1 .... \lambda_m$.

From definition 3, it is intuitively deductible that, an xF for a given V^{Dim} is actually the *context* for the V^{Dim} question.

4.2 XDW Relationships

Typically, in an XDW model, for one *x*FACT, there exists one or more VDims. Let the total number of V^{Dim} be ***n***. Let ${}_x R_d$ denote the relationship between the *x*FACT and a VDim and ${}_d R_d$ between two dimensions. The relationship between *x*FACT xF and VDim V^{Dim}_k, may be denoted as:

$${}_x R_d{}^k = (xF, V^{Dim}_k) \quad (3)$$

where $0 < k \le n$.

Also, the hierarchical relationship (dimensional hierarchy) between two VDims, V_k^{Dim} and V_{k+1}^{Dim} can be shown as:

$$_dR_d^{\ k} = (V_k^{Dim}, V_{k+1}^{Dim}) \qquad (4)$$

Both $_xR_d$ and $_dR_d$ may fall into one or more of the dimension specific relationships type (and constraints) as: (a) aggregate dimension (minimum , maximum, count, average), (b) time-variant dimension, (c) subject-variant dimension and (d) aggregate-descriptive dimension.

These types of relationships may correspond to one or more of the dimensional (conceptual) operators or queries, as described in (Nassis et al. 2005b). They may be grouped into: (i) aggregate selection, (ii) aggregate sort/order, (iii) implicit/explicit joins and (iv) aggregate grouping.

Analogous to the views in the LVM, at the logical level, these dimensional (conceptual) operators or queries are transformed into W3C use case query context algorithms (Nassis et al. 2005b) and later at the document/instance level to language specific query expression, such as XQuery and/or SQL.

5 ILLUSTRATIVE CASE STUDY EXAMPLE

As an illustrative case study for this paper, we intent to gradually build a e-Sol case study described in (Chang et al. 2003; ITEC 2002; R.Rajugan et al. 2005) as a XDW conceptual and logical model for the purposes of archiving and analysing e-Sol data for the purpose of business planning and intelligence. We refer to the e-Sol XDW model as e-Sol-W. Given below is the extended description and requirements of the e-Sol-W for the purpose of building a data warehouse and data marts.

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 a 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 high level of abstraction. From a local stakeholder/partners' perspective, the XDW model solves some of the problems faced by e-Sol. But from a global perspective, where multiple stakeholders/partner systems are involved (i.e. collaborative partners, global customers, etc.) and there is a need to support e-Sol's global information demand, the role and scope of XDW has to evolve and a new global warehouse model is inevitable and an unfortunate reality.

To illustrate our concepts, we highlight a few, simplified XDW user requirements. We consider a simplified XDW model for archiving and analysing warehouse bookings, income and capacity for the warehouses in the e-Sol. Some of these requirements include (Fig. 2-3):

(i) Warehouse booking: Warehouse booking records by (a) customers, (b) companies and (c) collaborative partners, grouped by (i) year, (ii) month and (iii) by warehouse location or by region (e.g. Asia-Pacific, China etc.). This information may help to rate customers and to plan warehouse capacity around the world, for a given time of the year.

(ii) Warehouse Capacity: Sort warehouse usage (i.e. slack space and near-full capacity measure) by year, month and quarterly and individual Q1, Q2, Q3 and Q4 capacity measure for (a) warehouse by region, and (b) warehouse by country.

(iii) Warehouse Revenue: List warehouse revenue by (a) year, (b) month (c) quarterly and (d) individual Q1, Q2, Q3 and Q4 for (i) individual warehouses, and (b) warehouses by region.

In our research, we use UML as the OO modelling language to represent XDW conceptual level artefacts (Fig. 2-3). Here, we use of UML just as a modelling notation as it is easily understood and standardised. As we have stated in our work in LVM, other OO languages may also be used instead of UML. It should be noted here that, we do not discuss the modelling and design of the xFACT here and the Fig. 3 is given only for illustrative purpose. A detailed discussion on the modelling xFACT can be found in (R.Rajugan, Chang & Dillon 2005).

Example 1: For example, as shown in Fig. 2, in the e-Sol-W case study, a VDim hierarchy, where there exist inheritance relationships between the VDim `Quarterly_WarehouseCapacity` and the individual Q1, Q2, Q3 and Q4 warehouse capacities.

Here, to model and represent XDW concepts, we utilize UML stereotypes. We introduced a new UML stereotype called `<<VDim>>` to model the virtual dimensions at the XDW conceptual level. Analogous to conceptual views in the LVM, this stereotype is similar to a UML class notation with a defined set of attributes and methods. The set of methods here can have either: constructors (to construct a VDim) or manipulators (to manipulate the VDim attribute set). Similar to conceptual views, at the XDW conceptual level, VDims can have additional semantic relationships such as generalization, aggregation, association and these can be shown using standard UML notation. In addition to this, two VDims can also have <<construct>> relationships with dependencies. Similarly, an *x*FACT can be represented using the <<xFACT>>, as shown Fig. 2-3.

We have stated that semantically related conceptual views could be logically grouped together as grouping of classes into a subject area. Further, a new view-hierarchy and/or constructs can be added to include additional semantics for a given user requirement. In the XDW conceptual model, when a collection of similar or related *conceptual views* are logically grouped together, we called it grouped VDims (Fig. 2), implying that it satisfies one or more logically related user requirement(s).

In addition, we can also construct additional conceptual view hierarchies as shown in Fig. 2. These hierarchies may form additional structural or dependency relationships with existing conceptual views or view hierarchies (grouped VDims) as shown in Fig. 2. Thus, it is possible that a cluster of dimensional hierarchy/ies can be used to model a certain set of user requirement/s. Therefore we argue that, this aggregate aspect can give us enough abstraction and flexibility to design a user-centred XDW model.

In order to model a hierarchy of VDim and capture the logical grouping among them, we utilize the ***package*** construct in UML. Thus, a grouped VDim hierarchy as shown in Fig. 2 can be represented using the UML package notation. This in practice describes our logical grouping of conceptual views and their hierarchies. Thus, we utilize packages to model our connected dimensions (Fig. 2).

In Fig 2, we show our case study XDW model with *x*FACT and VDims connected via the <<construct>> stereotype. Also, following the arguments presented, we can show that, the *x*FACT (shown in Fig 2-3) can be grouped into one logical construct and can be shown in UML as one package.

Example 2: In the e-Sol-W example, as shown in Fig. 2, WMS_Warehouse is an *x*FACT (Fig. 3) represented using a stereotyped UML package.

Example 3: Also, in Fig 2, a VDim, xFACT, and the relationship between VDim and the xFACT are shown using <<VDim>>, <<xFACT>> and <<construct>> stereotypes.

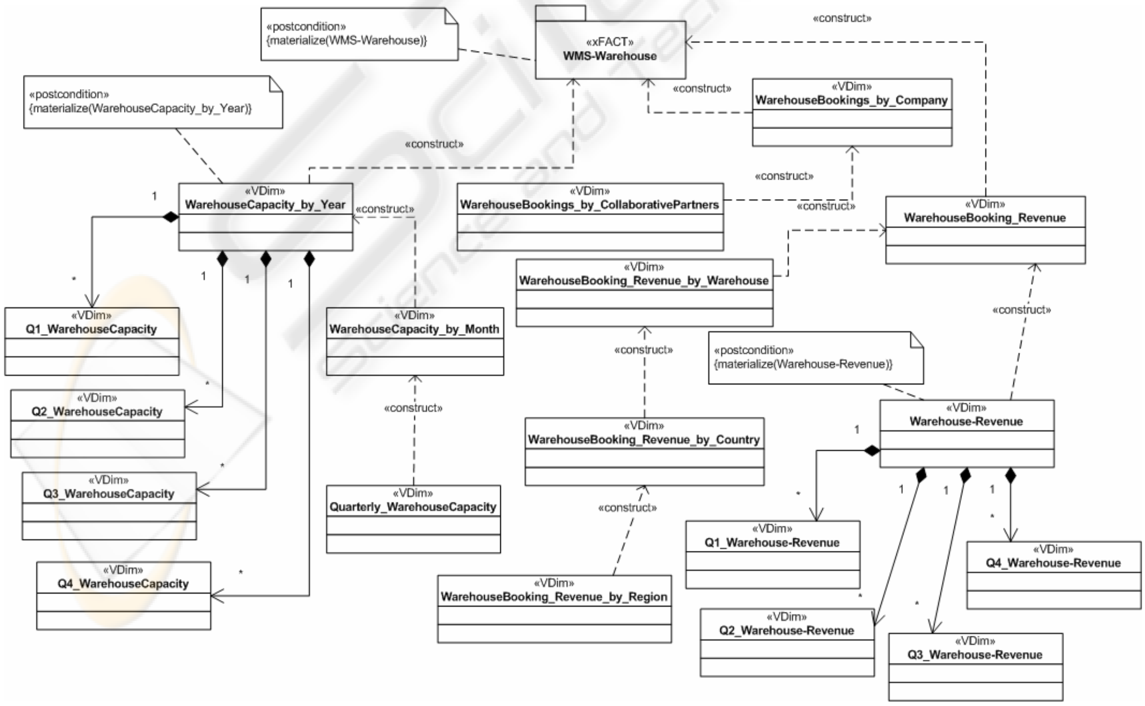

Figure 2: VDim hierarchy in e-Sol-W case study (with UML stereotypes).

Example 4: The VDim hierarchical relationships (Fig. 2) are shown using OO part-of (composition) relationships or using the `<<construct>>` stereotyped (dependency) relationships (i.e. view of a view).

Example 5: For example, as shown in Fig. 2, in the e-Sol-W case study, a VDim hierarchy, where there exist a part-of relationships between the VDim `Quarterly_WarehouseCapacity` and the individual Q1, Q2, Q3 and Q4 warehouse capacities.

Example 6: In e-Sol-W example, a conceptual views `Q1_Warehose-Revenue`, `Q2_Warehose-Revenue`, etc. are constructed in the given context of `Warehouse_Revenue` (Fig 2). The valid context is given by the e-Sol-W `WMS_Warehouse` objects (Fig 3).

Example 7: Also in the e-Sol-W example (Fig. 2), a conceptual views `Q1_WarehouseCapacity`, `Q2_WarehouseCapacity`, etc. are constructed in the given context of

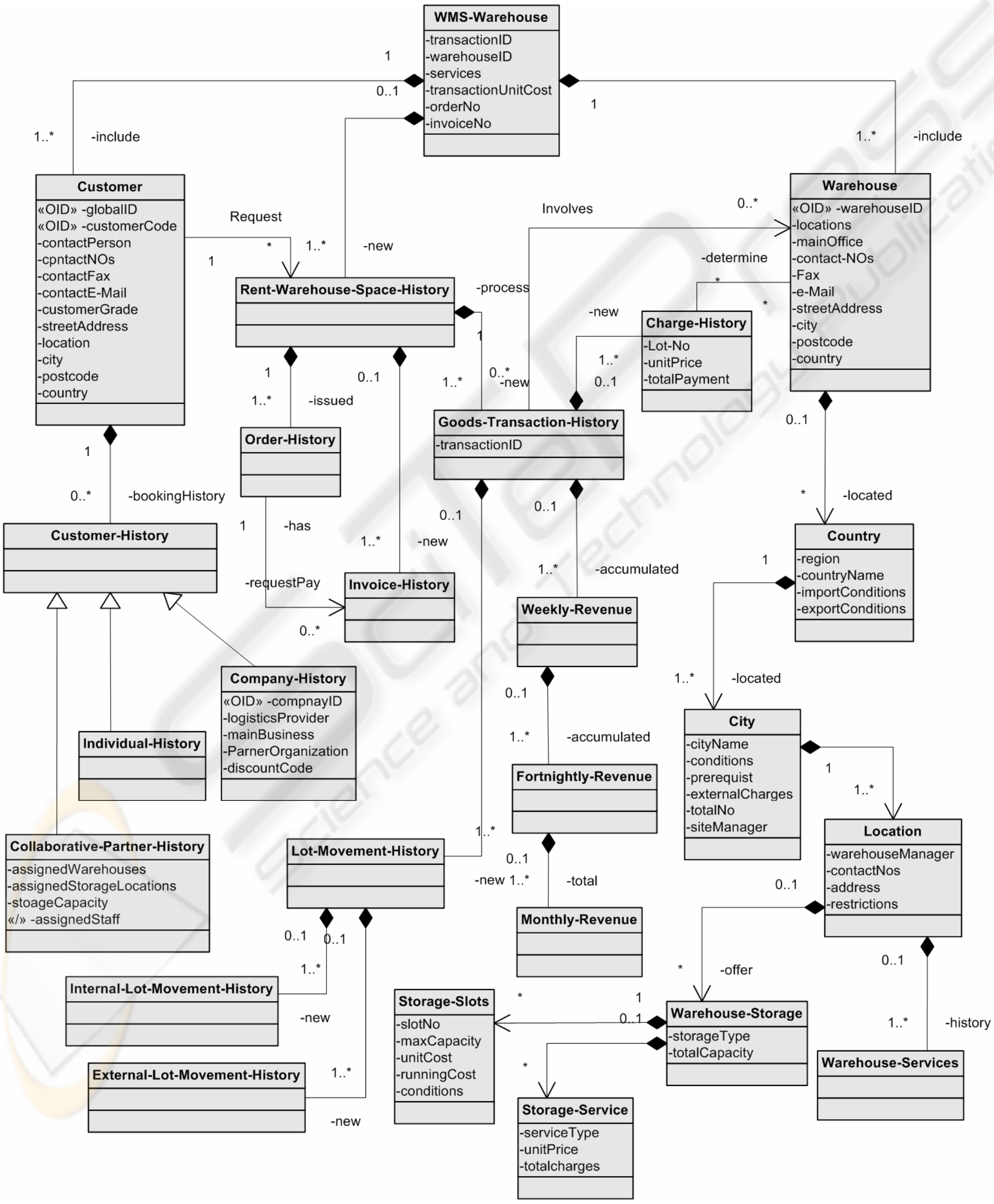

Figure 3: An xFACT example in the e-Sol-W (WMS-Warehouse).

`WarehouseCapacity_by_Year`. Again the valid context is given by the e-Sol-W `WMS_Warehouse` objects. Also, it should be noted that, due to the nature of xFACT, further dimensions may be constructed such as; `Regional-Warehouse-Capacity-by-Season`, `Warehouse-Capacity-by-Country`, etc. providing regional and/or global perspectives.

Example 8: In Fig. 2, in the xFACT is stated as a *materialized* at the conceptual level using OCL like syntax (R.Rajugan 2006).

Example 9: Similar to the above example, in the VDim hierarchy of, `WarehouseCapacity_by_Year` is stated as a *materialized* conceptual view (Fig. 2), implying that it is a persistence view (or VDim) during the lifetime of the system. VDim "`Warehouse-Revenue`" is also a materialized view.

6 CONCLUSION

In this paper, we presented an intuitive, a view-driven, conceptual framework (similar to the PIMs in MDA approach) to conceptually model, design and implement dimensions using the XML FACT repository in the XDW model.

For future work, some further issues deserve investigation. First, the investigation into OLAP support in the XDW model using such (virtual) dimensions. Second is the providing formal explanatory semantics to improve design and deployment of such dimensions using formalisms such as fuzzy sets. Finally, it is the formulation of a valid empirical study to consider and validate performance and quality issues using large datasets.

REFERENCES

Abelló, A, Samos, J & Saltor, F 2001, 'Understanding facts in a multidimensional object-oriented model', *4th Int. Workshop on DW and OLAP (DOLAP '01)*.

Chang, E, et al., T 2003, 'A Virtual Logistics Network and an e-Hub as a Competitive Approach for Small to Medium Size Companies', *2nd Human.Society@Internet Conf.*, Korea, pp. 265-71.

Dillon, TS & Tan, PL 1993, *Object-Oriented Conceptual Modeling*, Prentice Hall, Australia.

ECM-AIIM 2005, *The ECM Association*, AIIM, http://www.aiim.org.

Elmasri, R & Navathe, S 2004, *Fundamentals of database systems*, 4th edn, Pearson/Addison Wesley, New York.

Fankhauser, P & Klement, T 2003, 'XML for Data Warehousing Changes & Challenges', *DaWaK 2003*, Springer, Prague, pp. 1-3.

Feng, L & Dillon, TS 2003, 'Using Fuzzy Linguistic Representations to Provide Explanatory Semantics for Data Warehouses', *IEEE Trans. on Knowledge and Data Engineering (TOKDE)*, vol. 15(1), pp. 86-102

Giovinazzo, WA 2000, *Object-oriented data warehouse design : building a star schema*, Prentice Hall PTR, Prentice-Hall Int., Upper Saddle River, N.J. London.

Golfarelli, M, Rizzi, S & Vrdoljak, B 2001, 'Data warehouse design from XML sources', *Proc. of the 4th ACM Int. workshop on Data warehousing and OLAP*, ACM Press, Atlanta, Georgia, USA, pp. 40 - 7.

Gopalkrishnan, V, Li, Q & Karlapalem, K 1999, 'Star/Snow-flake Schema Driven Object-Relational Data Warehouse Design and Query Processing Strategies', *(DaWaK '99)*, Springer, Florence Italy.

Gray, P & Watson, HJ 1998, *Decision Support in The Data Warehouse*, Prentice Hall PTR, USA.

Gupta, A, Mumick, IS & (eds) 1999, *Materialized views: techniques, implementations, and applications*, eds A Gupta & IS Mumick, MIT Press.

Humphries, M, Hawkins, MW & Dy, MC 1999, *Data Warehousing: Architecture & Implementation*, Prentice Hall PTR, USA.

Inmon, WH, Imhoff, C & Battas, G 1996, *Building the operational data store*, John Wiley & Sons, NY, USA.

ITEC 2002, *iPower Logistics (http://www.logistics.cbs.curtin.edu.au/)*, viewed 2005.

Jeong, E & Hsu, C-N 2001, 'Introduction of Integrated View for XML Data with Heterogenous DTDs', *Proc. of the 10th Int. Conf. on Information and Knowledge Management (CIKM '01)*, ACM, US, pp. 151-8.

Kimball, R & Caserta, J 2004, *The data warehouse ETL toolkit : practical techniques for extracting, cleaning, conforming, and delivering data*, Wiley, Hoboken, NJ.

Kimball, R & Ross, M 2002, *The data warehouse toolkit : the complete guide to dimensional modeling*, 2nd edn, Wiley, New York.

Lucie-Xyleme 2001, 'Xyleme: A Dynamic Warehouse for XML Data of the Web', *IDEAS '01*, eds ME Adiba, C Collet & BC Desai, IEEE Computer Society 2001, Grenoble, France, pp. 3-7.

Luján-Mora, S & Trujillo, J 2004, 'A Data Warehouse Engineering Process', *Third ADVIS '04*, Springer-Verlag GmbH, Izmir, Turkey, p. 14.

Lujan-Mora, S, Trujillo, J & Song, I-Y 2002a, 'Extending the UML for Multidimensional Modeling', *Fifth Int. Conf. on the Unified Modeling Language and its applications (UML '02)*, Springer-Verlag London, UK, Dresden, Germany, pp. 290-304.

Lujan-Mora, et al., 2002b, 'Multidimensional Modeling with UML Package Diagrams', *Proc. of the 21st Int. Conf. on Conceptual Modeling (ER '02)*, Springer-Verlag UK, pp. 199-213.

Luján-Mora, et al., 2004, 'Advantages of UML for Multidimensional Modeling', *Proc. of the 6th Int. Conf. on ICEIS '04*, Porto, Portugal, April.

Luján-Mora, S, Vassiliadis, P & Trujillo, J 2004, 'Data Mapping Diagrams for Data Warehouse Design with UML', *23rd Int. Conf. on Conceptual Modeling (ER '04)*, vol. 3288, Shanghai, China, p. 191.

Medina, E, et al., J 2002, 'Handling Conceptual Multidimensional Models Using XML through DTDs', *Proc. of the 19th British National Conf. on Databases: Advances in Dbs*, vol. 2405, Springer, pp. 66 - 9.

Mohammed, S 2001, 'Object-Relational Data Warehouse', Master by Coursework (Major Thesis) thesis, La Trobe University, Melbourne, Australia.

Mohania, MK, Karlapalem, K & Kambayashi, Y 1999, 'Data Warehouse Design and Maintenance through View Normalization', *10th Int. DEXA '99*, vol. 1677, Springer, Florence, Italy, pp. 747-50.

Nassis, V, Dillon, TS, R.Rajugan & Rahayu, W 2006, 'An XML Document Warehouse Model', *The 11th Int. Conf. on Database Systems for Advanced Applications (DASFAA '06)*, Springer, Singapore, pp. 513-29.

Nassis, V, Dillon, TS, Rahayu, W & R.Rajugan 2006, 'Goal-Oriented Requirement Engineering for XML Document Warehouses', in J Darmont & O Boussaid (eds), *Processing and Managing Complex Data for Decision Support*, Idea Group Publishing, pp. 28-62.

Nassis, V, R.Rajugan, Dillon, TS & Rahayu, JW 2005a, 'A Requirement Engineering Approach for Designing XML-View Driven, XML Document Warehouses', *COMPSAC '05*, IEEE CS, Scotland, pp. 388-95.

Nassis, V, R.Rajugan, Dillon, TS & Rahayu, W 2005b, 'Conceptual and Systematic Design Approach for XML Document Warehouses', *Int. Journal of Data Warehousing and Mining*, vol. 1(3), no. 3, pp. 63-87

Nunamaker, JF, Chen, JM & Purdin, TDM 1991, 'System Development in Information Systems Research', *Journal of Mgmt. of IS*, vol. 7(3), pp. 89-106

OMG-CWM 2001, *The Common Warehouse Metamodel (http://www.omg.org/technology/cwm/)*, OMG, 2005.

OMG-MDA 2003, *The Architecture of Choice for a Changing World®, MDA Guide Version 1.0.1 (http://www.omg.org/mda/)*, OMG, 2005.

OMG-MOF™ 2003, *Meta-Object Facility (MOF™), 1.4 (http://www.omg.org/technology/documents/modeling_spec_catalog.htm#MOF)*, OMG, 2005.

Pokorn'y, J 2002, 'XML Data Warehouse: Modelling and Querying', *Proc. of the Baltic Conf. (BalticDB-IS '02)*.

R.Rajugan 2006, 'A Layered View Model for XML with Conceptual and Logical Extension, and its Applications', PhD thesis, University of Technology, Sydney (UTS), Australia, Sydney.

R.Rajugan, Chang, E & Dillon, TS 2005, 'Conceptual Design of an XML-View Driven, Global XML FACT Repository for XML Document Warehouses', *1st Int. Workshop on Data Management in Global Data Repositories (GRep '05), held in conjunction DEXA '05*, IEEE CS, Copenhagen, Denmark, pp. 1139 - 44.

R.Rajugan, Chang, E, Dillon, TS & Feng, L 2005, 'A Three-Layered XML View Model: A Practical Approach', *24th Int. Conf. on Conceptual Modeling (ER '05)*, Springer-Verlag, Austria, pp. 79-95.

R.Rajugan, et al., 2006, 'Modeling Views in the Layered View Model for XML Using UML', *Int. Journal of Web Information Systems (IJWIS), Troubador Publisher Ltd.*, vol. 2(2), pp. 95-117, June 2006.

Rahayu, W, Dillon, TS, Mohammed, S & Taniar, D 2001, 'Object-Relational Star Schemas', *13th IASTED Int. PDCS '01*, IASTED, LA, USA.

Theodoratos, D & Sellis, T 1999, 'Dynamic Data Warehouse Design', *1st DaWak '99*, Springer, Italy, pp. 1-10.

Trujillo, J, et al., 2003, 'Applying UML For Designing Multidimensional Databases And OLAP Applications. 13-36', *Advanced Topics in Database Research, Idea Group Publication*, vol. 2, pp. 13-36

Trujillo, J, Palomar, M, Gomez, J & Song, I-Y 2001, 'Designing Data Warehouses with OO Conceptual Models', *IEEE Computer Society, "Computer"*, December, 2001, pp. 66-75.

W3C-XML 2004, *Extensible Markup Language (XML) 1.0, (http://www.w3.org/XML/)*, The World Wide Web Consortium (W3C).

Xyleme 2001, *Xyleme Project (http://www.xyleme.com/)*.