Modeling Views for Semantic Web Using eXtensible Semantic (XSemantic) Nets

Rajugan, R.[1], Elizabeth Chang[2], Ling Feng[3] and Tharam S. Dillon[1]

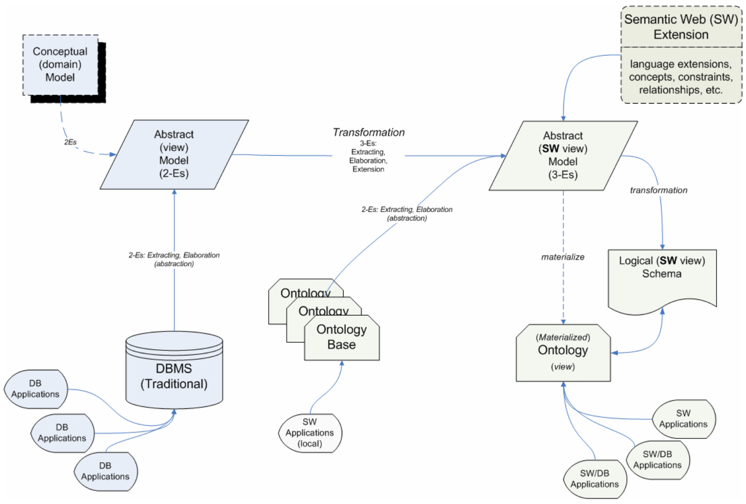
[1] eXel Lab, Faculty of IT, University of Technology, Sydney, Australia
{rajugan, tharam}@it.uts.edu.au
http://exel.it.uts.edu.au
[2] School of Information Systems, Curtin University of Technology, Australia
Elizabeth.Chang@cbs.cutin.edu.au
[3] Faculty of Computer Science, University of Twente, The Netherlands
ling@ewi.utwente.nl

Abstract. The emergence of Semantic Web (SW) and the related technologies promise to make the web a meaningful experience. Yet, high level modeling, design and querying techniques proves to be a challenging task for organizations that are hoping utilize the SW paradigm for their industrial applications, which are still using traditional database techniques. To address such an issue, in this paper, we propose a view model for the SW (SW-View), to SW-enable traditional solutions. First we outline the view model, its properties and some modeling issues, followed by some discussions on modeling such views (at the conceptual level). We also provide a brief discussion on how this view model is utilized in the design and construction of materialized ontology views to support extraction of sub-ontologies.

1 Introduction

Many traditional database concepts and techniques have been transformed and adopted to new web application platforms, which are mainly based on core Object-Oriented (OO) principles. For example, works such as [2-4, 16, 25] are good examples in this direction. The emergence of Semantic Web (SW) [33] and the related technologies promise to make the web a meaningful experience and it is another step towards the next generation of Enterprise Information Systems (EIS). However, success of SW and its applications heavily depends on utilization and interoperability of well formulated Ontology bases (and traditional data) in an automated, heterogeneous environment. For example, utilization, integration and extraction of ontology bases in the context of EIS, where, enterprise vocabularies can be automatically extracted from various distributed sources and be used in one or more SW (or traditional) applications and e-services. One such scenario is shown in Fig. 1.

This creates the need investigate successful database technologies, such as views, in the context of SW, where (materialized) ontology views [37] can be used for; (a) ontology extraction, (b) ontology versioning (c) SW-enabling traditional data sources and (d) sub-ontology generation, in an industrial settings. However, unlike traditional database systems, high level modeling, design and querying techniques still proves to be a challenging task for SW paradigm. This is mainly due to the nature of ontology bases and views, where, definitions and querying have to be done at high-level abstraction [30, 37]. Such a high-level view models can also be utilized in SW paradigm and also support and co-exist with existing traditional database architecture and/or enterprise transactional systems. A detailed discussion on the differences between the traditional database and SW technologies can be found in our work [26].



Semantic Web (SW) Extension
language extensions, concepts, constraints, relationships, etc.
Conceptual (domain) Model
2Es
Abstract (view) Model (2-Es)
Transformation
3-Es: Extracting, Elaboration, Extension
Abstract (SW view) Model (3-Es)
transformation
2-Es: Extracting, Elaboration (abstraction)
2-Es: Extracting, Elaboration (abstraction)
materialize
Logical (SW view) Schema
Ontology
Ontology
Ontology Base
(Materialized) Ontology (view)
DBMS (Traditional)
DB Applications
DB Applications
DB Applications
SW Applications (local)
SW Applications
SW/DB Applications
SW/DB Applications

Fig. 1. Databases and Ontology bases in EIS architecture (context diagram)

Conversely, Semantic Web directives are still at its infancy in areas such as data organization, meta-data models and query languages. But there is an exponential growth in new research directions in SW applications. These applications range from SW enabled traditional enterprise meta-data repositories (Fig. 1) to time-critical medical information and infectious decease classification databases. For such vast ontology bases to be successful in a distributed environment, the preliminary design and engineering of such ontology bases should follow a strict software engineering discipline [28]. Furthermore, supporting technologies for ontology engineering such as data extraction, integration and organization have be matured to provide adequate modeling and design mechanism to build, implement and maintain successful ontology bases. For such purpose, Object-Oriented (OO) paradigm seems to be an ideal choice as it has been proven in many other complex applications and domains [12, 18].

To address such an issue, in this paper, we propose a view model for SW, to support ontology views (Fig. 1). In contrast to SW language specific views (e.g. RDF [32] /RDF-S), the proposed view model is defined using a high-level modeling OO language that is capable of modeling ontologies and sub-ontologies (for e.g. XSemantic nets [14] or OMG's UML [23] or Ontology Web Language (OWL) [31]). Our main aim here is to "re-use" and "share" of view definitions among multiple implementation paradigms and frameworks (Fig. 1), namely; (a) (traditional) database systems, (b) database applications and (c) SW applications, as only the use, encoding, relationships and meta-data may change between these platforms in (a) to (c) above (*see* the example case study description in section 4). Thus we provide view definitions at the highest level of abstraction (i.e. conceptual level) which enables us to transform and map one view definition to a specific platform (i.e. (a), (b) or (c)), at the required level of abstraction (i.e. conceptual, schema or instance). Here, our focus is mainly on SW paradigm.

The rest of this paper is organized as follows. In section 2, we present some of the early work done in Semantic Web view models. Section 3 describes our work (including a brief outline on how our view model is applied in the MOVE system), followed by section 4, where we present an illustrative case study example to highlight some of our view model characteristics. Section 5 concludes the paper with some discussion on our future research directions.

2 Related Work

We can group the existing view models into four categories, namely; (a) classical (or relational) views [11, 13], (b) Object-Oriented (OO) view models [5, 22], (c) semi-structured (namely XML) view models [1, 10, 25] and (d) view models for SW. An extensive set of literature can be found in both academic and industry forums in relation to various view related issues such as (i) models, (ii) design, (iii) performance, (iv) automation and (v) turning/refinement, mainly supporting the 2-Es; data Extraction and Elaboration (with and some research directions towards 3-Es, i.e. 2-Es and data Extension). A comprehensive discussion on existing view models can be also found in [25, 26]. Here, we focus only on view models for SW.

In related work in Semantic Web (SW) [34] paradigm, some work has been done in views for SW [29, 30], where the authors proposed a view formalism for RDF document with support for RDF [32] schema (using a RDF schema supported query language called RQL). This is one of the early works focused purely on RDF/SW paradigm and has sufficient support for logical modeling of RDF views. The extension of this work (and other related projects) can be found at [21]. RDF is an object-attribute-value triple, where it implies object has an attribute with a value [15]. It only makes intentional semantics and not data modeling semantics. Therefore, unlike views for XML, views for such RDF (both logical and concrete) have no tangible scope outside its domain. In related area of research, the authors of the work propose a logical view formalism for ontology [36, 37] with limited support for conceptual extensions, where materialized ontology views are derived from conceptual/abstract view extensions.

Another area that is currently under development is the view formalism for SW Meta languages such as OWL. In some SW communities, OWL is considered to be a conceptual modeling language for modeling ontologies, while some others consider it to be a crossover language with rich conceptual semantics and RDF like schema structures [36]. It is outside the scope of this paper to provide argument for or against OWL being a conceptual modeling language. Here, we only highlight one of view formalism that is under development for OWL, namely views for OWL in the "User Oriented Hybrid ontology Development Environments" [19] project.

3 Our Work: SW-View Model

In this paper, we propose a layered view model for the SW paradigm (SW-view). Initially, we proposed a layered view model in our work for semi-structured data (namely XML) [25], and here we extend the model for SW paradigm.

In work with XML, we provided clear distinction between conceptual, logical and document levels views, as in the case of data engineering, there exists a need to clearly distinguish these levels of abstractions. But in the case of SW (e.g. ontologies), though there exists a clear distinction between conceptual and logical models/schemas, the distinction between the logical (or schema) level and document (or instance) level trends to overlap due to the nature of ontology bases, where concepts, relationships and values may present mixed sorts such as schemas and values [38].

Therefore, in the SW-view model, we provide a clear distinction between conceptual and logical views, but depending on the application, we allow an overlap between logical and document views. This is one of the main differences between the XML views and the SW-views. To our knowledge, other than our work, there exist no research directions that explore the conceptual and logical view formalism for the Semantic Web (SW) paradigm. This notion of SW-view model has explicit constraints and an extended set of conceptual operators to support ontology Extraction Methodology (OEM) [35, 37, 38].

3.1 Conceptual Views

In the layered view model, the conceptual views are views that are defined at the conceptual level with conceptual level semantics using a higher-level modeling languages such as UML [23] or XSemantic nets [14, 27]. Here, we use XSemantic nets. To understand the SW-view and its application in constructing ontology views, it is imperative to understand its concept and its properties. It should be noted here that, though there can be more elaborated definitions are possible depending on the application domain, here we provide a simplified generic conceptual view definition that can be easily applied.

***Definition 1*: A conceptual view** $\boldsymbol{V^c}$ is a 4-ary tuple $\boldsymbol{V^c} = (V^c_{name}, V^c_{obj}, V^c_{rel}, V^c_{constraint})$, where V^c_{name} is the name of the XML conceptual view V^c, V^c_{obj} is a set of objects in V^c, V^c_{rel} is a set of object relationships in V^c, and $V^c_{constraint}$ is a set of constraints associated with V^c_{obj} and V^c_{rel} 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 λ be a set of conceptual operators. $\boldsymbol{V^c} = (V^c_{name}, V^c_{obj}, V^c_{rel}, V^c_{constraint})$ is called a *valid conceptual view of the context* ***C, if and only if the following conditions satisfy;

1. For any object $\forall o \in V^{c}_{obj}$, there exist objects $\exists o_1, \ldots, o_n \in C_{obj}$, such that $o = \lambda_{1 \ldots}\lambda_m$ $(o_{1, \ldots,} o_n)$ where $\lambda_{1 \ldots}\lambda_m \in \tilde{\lambda}$. That is, o is a newly derived object from existing objects $o_1, \ldots, o_n$ in the context via a series of conceptual operators [24, 38] $\lambda_{1, \ldots}\lambda_m$ like select, join, etc.
2. For any constraint $\forall c \in V^{c}_{constraint}$, there exists a constraint $\exists c' \in C_{constraint}$ or a new constraint c'' constraints associated with V^{c}_{obj} or V_{rel}.
3. For any hierarchical relationship $\forall r^h \in V^{c}_{rel}$, there *does not exist* a relationship between one or more and V^{c}_{obj} and C_{obj}.
4. For any association relationship/dependency relationships $\forall r^a \in V^{c}_{rel}$, there *may exist a relationship between one or more* V^{c}_{obj} *and* C_{obj}.

The term ***context*** refers to the domain that interests an organization as a whole. It implies a meaningful collection of objects (or concepts), relationships (both structural and semantic) among these objects, as well as some constraints associated with the objects and their relationships, which are relevant to its applications. The following sections briefly address some of the unique characteristics of conceptual views; (i) conceptual operators, (ii) some modeling issues and (iii) the descriptive constraint specification for conceptual views using XSemantic nets.

3.2 Conceptual Operators

A Context is presented in XSemantic nets using modeling primitives like object (node), attribute (simple node), relationship (directed edges) and constraint in this study. To enable the construction of a valid conceptual view from a context, we introduced the notion of *conceptual operator* ($\tilde{\lambda}$)[24]. These operators are grouped into set operators, namely union, difference, intersection, Cartesian product and unary operators namely projection, rename, restructure, selection and joins, and can facilitate systematic construction of conceptual views from context. These conceptual operators can be easily transformed into query segments, user-defined functions and/or procedures for implementation. By doing so, they help the modeler to capture view construct at the abstract level without knowing or worrying about query/language syntax. The set of binary and unary operators provided here is a complete or basic set; i.e. other operators, such as division operator and compression operator [38] can be derived from these basic set of operators.

3.3 Modeling Conceptual Views

In this paper, to model conceptual views, we use XSemantic nets. Other modeling notations used to model conceptual views can be found in [38]. XSemantic net provides a well defined, rich semantics to visually model a given domain into needed level of abstraction [14]. In the case of Ontology engineering, XSemantic nets provide rich collection of OO concepts and elements, namely; (i) classes (similar to concepts in ontology), (ii) attributes (iii) relationships (and cardinality constraints) between classes, (iv) relationships (and cardinality constraints) between class and its attributes and (v) a rich set of constraints (*see* section 3.4 below). Some of the XSemantic net notations are given Fig. 2 and an illustrative case study example model is given in section 4.

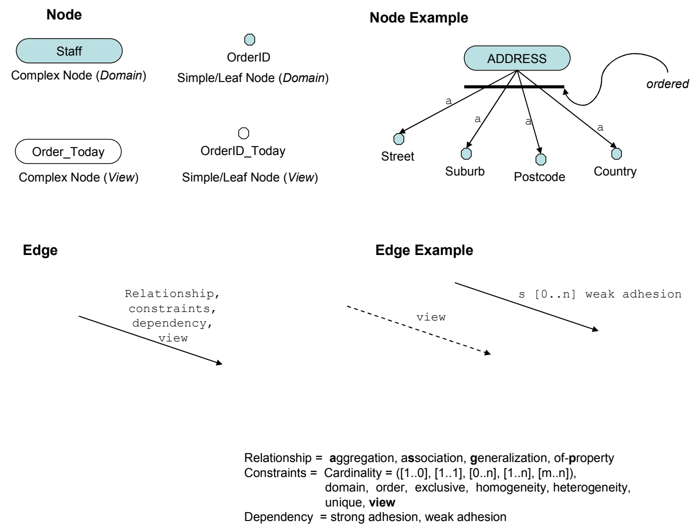

Node
Staff
Complex Node (*Domain*)
OrderID
Simple/Leaf Node (*Domain*)
Order_Today
Complex Node (*View*)
OrderID_Today
Simple/Leaf Node (*View*)
Node Example
ADDRESS
ordered
a
a
a
a
Street
Suburb
Postcode
Country
Edge
Relationship, constraints, dependency, view
Edge Example
view
s [0..n] weak adhesion
Relationship = **a**ggregation, **as**sociation, **g**eneralization, of-**p**roperty
Constraints = Cardinality = ([1..0], [1..1], [0..n], [1..n], [m..n]), domain, order, exclusive, homogeneity, heterogeneity, unique, **view**
Dependency = strong adhesion, weak adhesion

Fig. 2. XSemantic net notations

Base on the V^c definition 1 above, in XSemantic nets, $V^c{}_{obj}$ are shown using (simple/complex) nodes, $V^c{}_{rel}$ using edges and $V^c{}_{constraint}$ using constraints defined over (a set of) node/(s) and (a set of) edges.

3.4 Modeling Conceptual View Constraints

One of the main differences between traditional data modeling and modeling ontologies, is the constraint specification. In ontology modeling, it requires a rich set of high-level constraint specification mechanism. In the case of views (both traditional data and ontology views), it is usually specified by the data/query language in which they are defined. For example, in relational model, views are defined using SQL and a limited set of constraints can be defined using SQL[11, 13], while in Object-Relational and OO models, views have similar constraints but they are more extensive and explicit due to the nature of the data model. The views here are constructed and specified by using DBMS specific (such as OQL[7]) and/or external languages (such as C++, Java or O_2C[5]). It is a similar situation in views for semi-structured data paradigm, where rich set of view constrains are defined using languages such as OQL based LOREL [6, 17]. Today, in the case of Ontology engineering (and in ontology views), this is still holds true, where constraints are specified using programming modules than at the schemata and/or logical level. In doing so, the constraints are implicit and mostly accessible only at runtime of the system and not at the modeling and/or design time.

But the work by authors of [10] provides some form of higher-level view constraints (under ORA-SS model) for XML views, while the work in [30] provides some form of logical level view constraints to be defined in views for in SW/RDF paradigm. Here, for our view formalism, we look into using XSemantic net as the (visual) constraint specification language.

3.5 Constraint Specification Using XSemantic Net

XSemantic net is a modified semantic network model, to model data domains under the OO paradigm [14, 27]. In XSemantic nets, due to its structural similarity to semi-structured data (e.g. XML, RDF etc.), most data/schema specific constraints are build-in to the model. There exist no need to have additional (textual) constraint specification language (e.g. such as OCL). These constraints are grouped into three categories [14], namely; (a) constraints over an edge, (b) constraints over a set of edges and (c) constraints over an edge. In addition, further constraints can be defined for conceptual views including; (i) domain constraints (range of values, min, max, pattern etc), (ii) constructional contents (set, sequence, bag, ordered-set), (iii) ordering (iv) explicit homogenous composition/heterogeneous compositions, (v) adhesion and/or dependencies (vi) exclusive disjunction and many more. Specifying these constraints in XSemantic nets (or UML/OCL) for conceptual views is similar to that of stored domain object constraints. The notations used in XSemantic net are given in Fig. 2. In section 4, we demonstrate constraint specification using XSemantic nets using some case study examples.

3.6 Conceptual Views on the MOVE System

Here, we briefly discuss how SW-views can be applied in the Materialized Ontology View Extractor (MOVE) system [36] for ontology extraction. The MOVE system was initially proposed by Wouters et al. [35-37], for the construction of *optimized* materialised ontology views, with emphasis on automation and quality of the views generated. The MOVE view process includes model and design of conceptual views with the utilization of restricted conceptual operators in deriving materialized ontology views. Some of the restricted view operators (derived from one or more SW-view conceptual operators) include [37, 38]; (a) synonymous rename (2) selection and (3) compression. A detailed discussion in this topic can be found in [38] and detailed work on MOVE can be found in [35-37].

Definition 3**:** [38](Informal) A Strict Semantic Web View (or Ontology View) is a materialized SW-view that is derived from an ontology (called the base ontology). The derivation can consist of any (combination) of the following operations; synonymous rename, selection and compression.

4 An Illustrative Case Study Example

To help illustrate our concepts, we conduct a real-world case study in a fictitious global logistic company called LWC & e-Solutions Inc., e-Sol in short. The e-Sol Inc. aims to provide logistics, warehouse, and cold storage space for its global 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 [8] 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 [8, 9, 20]. Here, we use this system as the base to model and integrate (using views) various (traditional) databases, ontology bases and other sub-ontology vocabularies used at various customer and collaborative partner locations (Fig. 1).

In e-Sol, due to the business process, data semantics have to be in different formats (ontology bases, databases, XML and vocabularies) to support multiple systems, customers, warehouses and logistics providers. Also, data have to be duplicated at various points in time, in multiple databases, to support collaborative business needs. In addition, since new customers/providers join the system (or leave), the data formats has to be dynamic and should be efficiently duplicated without loss of semantics. This presents an opportunity to investigate how to integrate and utilize various customers' and collaborative partners' (data and ontology) bases for mutual benefit and for SW applications. The following examples highlight some of the conceptual views developed for the e-Sol. Note: It should be note that, the examples and the figures given for the e-Sol are demonstration purpose only and do not provide the complete ontology base model of the system.

Example 1: "`staff`", "`order`", and "`customer`" can be some of the context examples in the e-Sol system.

Example 2: "`processed-order`" and "`overdue-order`" are two contrasting conceptual views in the context of "`order`" of the e-Sol system.

Example 3: "`Warehouse-Manager`" is a valid conceptual view, named in the context of "`Staff`". It is constructed using the conceptual SELECT operator [24], which can be shown as;

$$\sigma_{\texttt{warehouse-Staff.Role="manager"}}(\texttt{Users}).$$

Example 4: Similarly, the conceptual view of name "`Site-Manager`" in the given context "`Staff`".

Example 5: In the case of conceptual view "`Warehouse-Manager`" (Fig. 3), we indicate the unique `staffID` using the unique constraint.

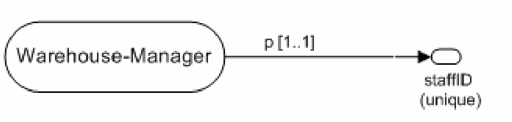


Fig. 3. Unique Constraint

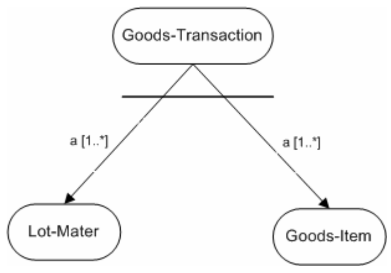


Fig. 4. Ordered composition in e-Sol.

Example 6: In real-world, composite objects being in an aggregation with one or more sub-objects (Fig. 4), they also can be in a pre-defined order. This signifies an important OO concept, *ordered composition*.

Example 7: In the case of conceptual views "`Lot-Movement`", the exclusive disjunction between `Internal-Lot-Movement` (stored goods change owners) and `External-Lot-Movement` (goods shipped outside the warehouse) can be shown as in Fig. 5.

Example 8: *One* `Goods-Type` composes of *one-or-more* `Goods-Sub-Type`/(s) and one `Goods-Item` is associated with *one-or-more* `Goods-Sub-Type`/(s), as shown as in Fig. 6.

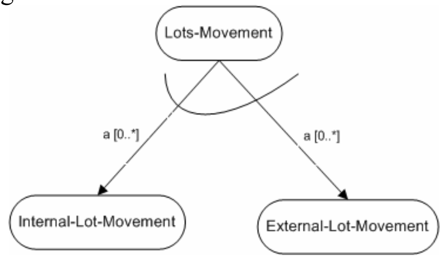


Fig. 5. Exclusive disjunction constraint

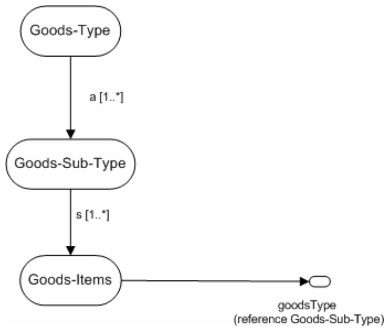


Fig. 6. A cardinality constraint example

Example 9: In the case of conceptual views "`Warehouse-Manager`" and "`Warehouse-Staff`", in the context of "`Staff`", we indicate the adhesion relationship, as shown as in Fig. 7.

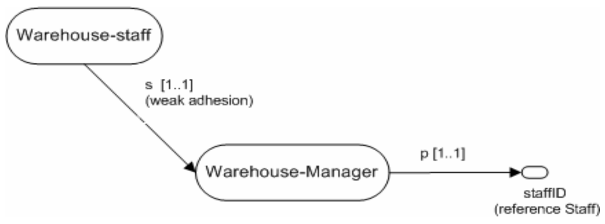


Fig. 7. Dependency / adhesion constraint

Example 10: In the case of conceptual view "`Income`", the dependency (constraint) relationships shown in Fig. 8 hold true.

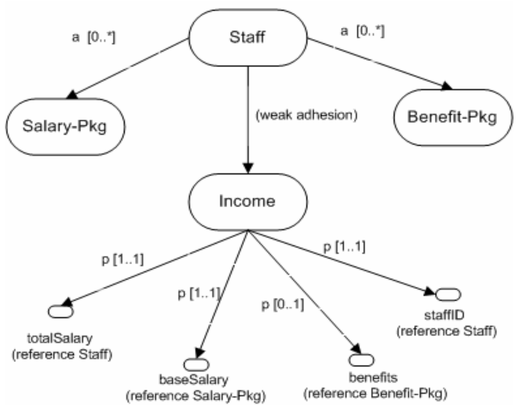


Fig. 8. A complex dependency constraint

Example 11: A compression of elements indicates that those elements are replaced by a single element in the ontology view [38]. The element itself can be a new element, but it will not provide additional semantic information (compared to the base ontology). The compression operator constituted of one or more of unary operations combined in sequence.

5 Conclusion and Future Work

Views have proven to be very useful in databases and here, we discussion on an abstract view model for SW (SW-view). First, we described the opportunities and challenges for utilizing SW technologies for EIS and databases. Then we briefly provided some arguments for our SW-view model and discussed its properties, definitions and modelling aspects. We also briefly showed how such view model is applied in the MOVE system for ontology extraction. Finally, we presented a practical walkthrough of the view model using an industrial case study example.

For future work, some further issues deserve investigation. First, the investigation of a formal mapping approach to conceptual view constraints, to automate the view constraint model transformation between the SW-view model and one or more SW language (such as RDF and OWL schema) constraints. Second, the automation of the mapping process between conceptual operators to various SW (high-level) query language expressions (e.g. RDQL) with emphasis on performance.

References

1. S. Abiteboul, "On Views and XML," Proc. of the 18th ACM PODS '99, USA, 1999.
2. S. Abiteboul, et al., "Active Views for Electronic Commerce," Proc. of the 25th Int. Conf. on VLDB, Edinburgh, Scotland, 1999.
3. S. Abiteboul, O. Benjelloun, I. Manolescu, T. Milo, and R. Weber, "Active XML: A Data-Centric Perspective on Web Services," BDA, 2002.
4. S. Abiteboul, O. Benjelloun, I. Manolescu, T. Milo, and R. Weber, "Active XML: Peer-to-Peer Data and Web Services Integration," Proc. of the 28th Int. Conf. on VLDB, HK, China, 2002.
5. S. Abiteboul and A. Bonner, "Objects and Views," ACM SIGMOD Record, Proc. of the Int. Conf. on Management of Data (ACM SIGMOD '91), 1991.
6. S. Abiteboul, J. Quass, J. McHugh, J. Widom, and J. Wiener, "The Lorel Query Language for Semistructured Data," *Int. Journal on Digital Libraries*, vol. 1, pp. 68-88, 1997.
7. R. G. G. Cattell, et al., "The Object Data Standard: ODMG 3.0," Morgan Kaufmann, 2000, pp. 300.
8. 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.
9. E. Chang, et al., "Virtual Collaborative Logistics and B2B e-Comm.," e-Business Conf., NZ, 2001.
10. Y. B. Chen, T. W. Ling, and M. L. Lee, "Designing Valid XML Views," Proc. of the 21st Int. Conf. on Conceptual Modeling (ER '02), Tampere, Finland, 2002.
11. C. J. Date, *An introduction to database systems*, 8th ed. New York: Pearson/Addison Wesley, 2003.
12. T. S. Dillon and P. L. Tan, *Object-Oriented Conceptual Modeling*: Prentice Hall, Australia, 1993.
13. R. Elmasri and S. Navathe, *Fundamentals of database systems*, 4th ed. New York: Pearson/Addison Wesley, 2004.
14. 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.

15. 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.
16. L. Feng and T. S. Dillon, "An XML-Enabled Data Mining Query Language XML-DMQL," *Int. Journal of Business Intelligence and Data Mining*, 2005.
17. R. Goldman, J. McHugh, and J. Widom, "From Semistructured Data to XML: Migrating the Lore Data Model and Query Language," Proc. of the 2nd Int. Workshop on the Web and Databases (WebDB '99), Philadelphia, Pennsylvania, 1999.
18. I. Graham, A. C. Wills, and A. J. O'Callaghan, *Object-oriented methods : principles & practice*, 3rd ed. Harlow: Addison-Wesley, 2001.
19. HyOntUse, "User Oriented Hybrid Ontology Development Environments, (http://www.cs.man.ac.uk/mig/projects/current/hyontuse/)," 2003.
20. ITEC, "iPower Logistics (http://www.logistics.cbs.curtin.edu.au/)," 2002.
21. KAON, "KAON Project (http://kaon.semanticweb.org/Members/rvo/Folder.2002-08-22.1409/Module.2002-08-22.1426/view)," 2004.
22. W. Kim and W. Kelly, "Chapter 6: On View Support in Object-Oriented Database Systems," in *Modern Database Systems*: Addison-Wesley Publishing Company, 1995, pp. 108-129.
23. OMG-UML™, "UML 2.0 Final Adopted Specification (http://www.uml.org/#UML2.0)," 2003.
24. R.Rajugan, E. Chang, T. S. Dillon, and L. Feng, "A Layered View Model for XML Repositories & XML Data Warehouses," The 5th Int. Conf. on Computer and Information Technology (CIT '05), Shanghai, China, 2005.
25. R.Rajugan, E. Chang, T. S. Dillon, and L. Feng, "A Three-Layered XML View Model: A Practical Approach," 24th Int. Conf. on Conceptual Modeling (ER '05), Klagenfurt, Austria, 2005.
26. R.Rajugan, E. Chang, T. S. Dillon, L. Feng, and C. Wouters, "Modeling Ontology Views: An Abstract View Model for Semantic Web," 1st Int. IFIP/WG 12.5 Working Conf. on Industrial Applications of Semantic Web (IASW '05), Jyvaskyla, Finland, 2005.
27. R.Rajugan, 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.
28. P. Spyns, R. Meersman, and J. Mustafa, "Data Modeling Versus Ontology Engineering," SIGMOD, 2002.
29. R. Volz, D. Oberle, and R. Studer, "Implementing Views for Light-Weight Web Ontologies," Seventh Int. Database Engineering and Applications Symposium (IDEAS'03), Hong Kong, SAR, 2003.
30. R. Volz, D. Oberle, and R. Studer, "Views for light-weight Web ontologies," Proc. of the ACM Symposium on Applied Computing (SAC '03), USA, 2003.
31. W3C-OWL, "OWL: Web Ontology Language 1.0 reference (http://www.w3.org/2004/OWL/)," W3C, 2004.
32. W3C-RDF, "Resource Description Framework (RDF), (http://www.w3.org/RDF/)," 3 ed: The World Wide Web Consortium (W3C), 2004.
33. W3C-SW, "(http://www.w3.org/2001/sw/)," W3C, 2005.
34. W3C-SW, "Semantic Web, (http://www.w3.org/2001/sw/)," W3C, 2005.
35. C. Wouters, T. S. Dillon, J. W. Rahayu, and E. Chang, "A Practical Walkthrough of the Ontology Derivation Rules," Database and Expert Systems Applications : 13th Int. Conf. (DEXA '02), Aix-en-Provence, France, 2002.
36. C. Wouters, T. S. Dillon, J. W. Rahayu, E. Chang, and R. Meersman, "Ontologies on the MOVE," 9th Int. Conf. on Database Systems for Advanced Applications (DASFAA '04), Jeju Island, Korea, 2004.
37. C. Wouters, T. S. Dillon, J. W. Rahayu, E. Chang, and R. Meersman, "A Practical Approach to the Derivation of a Materialized Ontology View," in *Web Information Systems*,. USA: Idea Group Publishing, 2004.
38. C. Wouters, R.Rajugan, T. S. Dillon, and J. W. Rahayu, "Ontology Extraction Using Views for Semantic Web," in *Web Semantics and Ontology*, USA: Idea Group Publishing, 2005.