

Subject-Oriented Semantic Knowledge Warehouse (SSKW) to Support Cognitive DSS

Tasneem Memon, Jie Lu, Farookh Khadeer Hussain, and Rajan Rauniyar

Decision Systems & e-Service Intelligence Laboratory (DeSI),
Centre for Quantum Computation & Intelligent Systems (QCIS)
University of Technology Sydney, Australia
{tasneem.memon,jie.lu,farookh.hussain}@uts.edu.au,
rajanrauniyar@gmail.com

Abstract. The communication between cognitive DSS and data warehouse tends to be inefficient due to their contradictory knowledge/data oriented nature. Data-to-knowledge conversion requires specialized techniques, whereas knowledge-to-data conversion results in *loss of knowledge*. To address these issues, a *subject-oriented semantic knowledge warehouse (SSKW)* is proposed, to provide *relevant* and *precise knowledge* to CDSS. The SSKW consists of: a) *object/process/event/relationship (OPER)* model to store domain knowledge in a unified fashion; and, b) a *subjective view* database, containing opinions of stakeholders about various OPER knowledge elements. A case study to compare the performance of the SSKW-based CDSS against a DW-based CDSS is presented. The results show that SSKW improves communication efficiency, provides *relevant* and *precise domain knowledge* to CDSS in less decision cycles, minimizes the *loss of knowledge*, and helps decision maker to quickly grasp the decision situation through its *human-centric* nature.

Keywords: Knowledge warehousing, knowledge representation, cognitive decision support, business intelligence.

1 Introduction

Despite *knowledge* being the key factor to gain competitive advantage, BI technologies, especially data warehouses (DW), do not support handling of *knowledge* [1–4]. DW filter out a great deal of *knowledge* during the *loading phase*, resulting in the loss of crucial *decision making knowledge* [5]. The field of CDSS, on the other hand, has made significant progress regarding *knowledge* processing; yet CDSS has to rely upon DW for back-end [6–9]. This poses two issues: 1) huge amounts of *knowledge* is lost during knowledge-to-data (CDSS-to-DW) conversion [2]. 2) *data mining* is essential to infer hidden patterns from data to create *knowledge* during data-to-knowledge (DW-to-CDSS) conversion. A knowledge-enabled back-end was thus deemed essential to support decision making, and *knowledge warehouses (KW)* were proposed [2, 4, 6]. KW are hybrid knowledge-base(KB) systems (collection of knowledge-bases) [2]. The main disadvantage of

hybrid-KB approach is knowledge integration overhead, which can decrease the performance with growing KBs [10]. It is argued here that a KW should store knowledge in a unified format. A *subject-oriented semantic knowledge warehouse (SSKW)*, is proposed to store *knowledge* in *unified, subject-oriented and semantic* manner. SSKW addresses the issues of *inefficient communication*, and the *loss of knowledge*. The paper is organized as follows. Section 2 presents literature review, followed by SSKW schema in Section 3. Section 4 describes the SSKW architecture and implementation. The case study is given in Section 5, followed by comparison of SSKW and DW, and conclusion.

2 Literature Review

DW provide decision makers with huge amounts of structured data, but fall short in handling *knowledge*, thus only partially supporting decision making process [3,7,11]. To fill this gap, *knowledge warehouses (KW)* were proposed [2,4], based on Nonaka's [1] *knowledge spiral model*. The term has been used for various applications [3]; however, Nemati et al [2] have defined it as a *novel technology* to collect, organize, store and disseminate *knowledge*. This paper is based on this characterization of KW. The KW by Nemati et al [2] is a collection of knowledge-bases (hybrid knowledge-base). Dymond [4] also proposed a similar hybrid-KB based KW. Several such KW have been designed [12–15]. Among these, ISYMOD [13] is the closest to the SSKW storage structure, while Pedersen [12] proposes a powerful tool for *evolution of knowledge*, i.e. *subjective opinions*, demonstrating the importance of individual's opinions.

The relational technology falls short in representing the complex and unstructured nature of *knowledge* [16,17]. The fact can be observed from a KW based on relational technology, by Levy et al [18,19]. The limitations of relational model hinder real-world object/relationship representation in this KW. Also, OLAP reports produced are lengthy and time-consuming [19]. Object-oriented approach on the other hand, has proven to be effective in representing real world entities in intuitive format, providing basis for rich semantics [20–22]. We use *object-oriented model* as the foundation to employ *semantics* and *subject-oriented approach* in the design of *SSKW*.

3 Subject-Oriented Semantic Knowledge Warehouse (SSKW) Schema

This section describes SSKW, and its components, *OPER model* and *SV database*.

3.1 Object/Process/Event/Relationship (OPER) Model

To define *standard knowledge* format, we categorized *knowledge* at syntactic level to create "a virtual level of knowledge representation where the dominant traits are not domain-dependent" [23]. After conducting a study, three fundamental

knowledge types were identified in a typical business environment: *objects*, *processes* and *events*. However, to simulate real-world, and the reasoning process of human mind, these elements must be linked in a meaningful way [24, 25]. Thus, *relationships* are added to them.

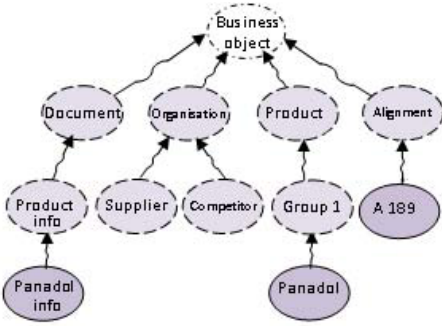


Fig. 1. Example of Object Hierarchy

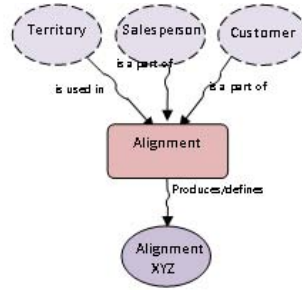


Fig. 2. Example of a Process

Object. *Object* defines an entity, such as *person*, *organisation* or *document*. The object hierarchy has three design levels (Fig. 1): 1) the abstract level, such as *business object*; 2) the sub-type level, such as *organisation*, which can span over multiple internal levels; and 3) the uniquely-identifiable level, such as *A189*. SSKW objects can have three types of properties; 1) *single-valued* such as name and age; 2) *objects*, such as organization; and 3) *processes* to define object’s behavior, such as *decision making*.

Process. It is a skeleton for business procedures, such as *delivery*, *sample checking* and *marketing*. It may involve one or more *objects*, in two categories: 1) Input objects, which include *Objects working* and *Objects being worked on*; and 2) Output objects (objects created) (Fig.2)

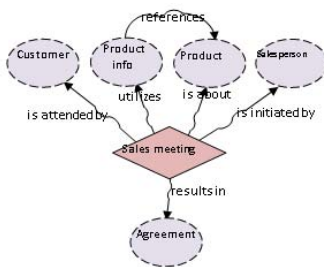


Fig. 3. Example of Event

Event. An *event* represents an occurrence, such as *meeting*, *cold call* and *presentation*. Event is a spatio-temporal unit, having *time* and *location* as its defining factors. One or more *objects/processes* may be involved. An event does not have an output 3.

Relationship. *Relationship* defines the nature of association between two *objects* (Fig. 1,2,3). The SSKW relationships are different than those in relational database where a relationship is applied to all the instances of an entity. That is, *instances*

of the same type must have the same set of relationships. In SSKW however, *objects* of the same type may hold different relationships. SSKW *relationships* may be cause-effect, implication, subtype, similar-to, instance and reverse [26].

3.2 Subjective Views

		OPER Knowledge element (KE)				
		KE ₁	KE ₂	KE ₃	...	KE _n
Customer	C ₁	SV ₁₁	SV ₁₂	SV ₁₃	...	SV _{1n}
	C ₂	SV ₂₁	SV ₂₂	SV ₂₃	...	SV _{2n}
	C ₃	SV ₃₁	SV ₃₂	SV ₃₃	...	SV _{3n}

	C _m	SV _{m1}	SV _{m2}	SV _{m3}	...	SV _{mn}

Fig. 4. Customers’ Subjective View

Based on *subject-oriented programming (SoP)* [27], *Subjective views (SV)* contain *opinions* and (*attitudes*) of stakeholders, such as customers, employees or competitors, about OPER knowledge elements, to assist decision makers in perceiving the situation from the perspective of stakeholders. A SV is a matrix, with OPER knowledge elements for one dimension, and the stakeholder-type (such as customers or managers) for the other. Fig.4 demonstrates a

customer SV, where *rows* represent *customers*, and *columns* represent *OPER knowledge elements* such as *product*. Each element in the matrix contains customer’s feedback about a specific knowledge element. This feedback can be a number (from likert scale), or text-based (from blogs). A SV matrix is used to calculate two values; a) the current trend for an OPER knowledge element, by aggregating the feedback of all the stakeholders; and b) the potential future preferences of a stakeholder, by studying their feedback about various OPER knowledge elements.

4 Subject-Oriented Semantic Knowledge Warehouse (SSKW) architecture and Implementation

This section gives an overview of SSKW conceptual model and implementation.

4.1 SSKW Architecture

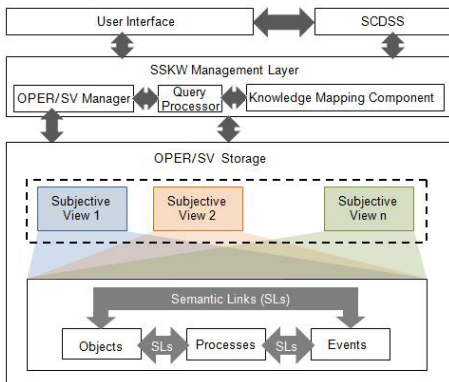


Fig. 5. The conceptual model of SSKW

The architecture consists of three main components; *OPER Storage*, the *SVs* and *Knowledge Management (KM) layer*. As discussed previously, OPER storage contains *objects*, *processes*, *events* and *semantic relationships* between them; while SVs contain the opinions of stakeholders about OPER knowledge elements. SSKW may contain more than one SVs (Fig 5). The *SSKW management layer* consists of the *OPER/SV manager* to interact with the storage, *query processor* to

manage the queries received from GUI, and *knowledge mapping component (KMC)* to maps OPER knowledge elements with their counterparts in CDSS. Following are the formal definitions of OPER and SV components.

Let K be the set of all knowledge elements (objects (O), processes(P), events(E) and relationships(R)) in a particular domain, i.e.

$$K = \{k_1, k_2, \dots, k_n \mid k \in O, P, E \text{ or } R\} \tag{1}$$

Where, O is the set of all the *objects* in the domain, P is the set of all *processes*, and E is the set of all *events*. Whereas *relationship* is defined as,

$$R = (m_{pre}, SL, m_{post}) \tag{2}$$

Here, m_{pre} and m_{post} are the knowledge elements of types O, P or E. SL is the set of all *semantic links* in the domain.

The complete set of *subjective views* is denoted by SV , and is defined as the set of matrices, in which each matrix represents *subjective views* of a particular group of stakeholders, about OPER knowledge elements. A SV is defined as follows:

$$SV = \{[sv_1], [sv_2], \dots, [sv_l]\} \tag{3}$$

Where, each $[sv]$ is a two dimensional matrix, as defined below:

$$[sv] \begin{matrix} & E_1 & E_2 & \dots & E_n \\ P_1 & \begin{bmatrix} sv_{11} & sv_{12} & \dots & sv_{1n} \\ sv_{21} & sv_{22} & \dots & sv_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ sv_{m1} & sv_{m2} & \dots & sv_{mn} \end{bmatrix} \\ P_2 & \\ \dots & \\ P_m & \end{matrix}$$

where $[sv]$ represents the sets of *views* of *participants (P)* from a particular group of stakeholders, such as *customer group* or *executive group*.

Each *entry (element)* of this matrix is mapped to exactly one knowledge element (E) in the set K of equation (1).

4.2 Implementation of SSKW

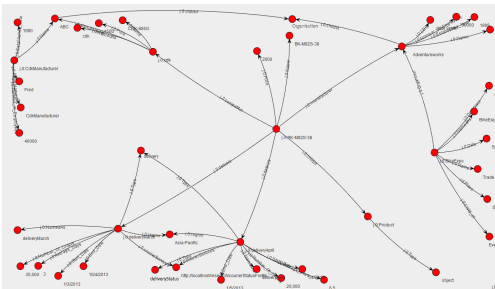


Fig. 6. Example of Knowledge storage in OPER

The SSKW architecture is implemented to test and evaluate its practicality and performance. The SVs use relational database, while OPER model is implemented using RDF. The main advantage of OPER storage is that unlike DW, in SSKW *knowledge* can be accessed through any dimension. As shown in Fig. 6, *BK-M82S-38* is

type *product*, linked with its *manufacturer AdventureWorks*, *competitor CDK-bike*, and two *delivery* processes, *deliveryMarch* and *deliveryApril* of type *delivery*. When *BK-M82S-38* is accessed, all their associated knowledge elements become available through *semantic relationships*. Similarly, accessing *delivery* provides direct access to all its child processes for all the products.

5 Case Study

We test our system against a DW-based CDSS, FACETS by Niu et al [7]. The decision situation is: "Why the sales of bike (*BK-M82S-38*) have dropped over the past 2 weeks?" FACETS solved this problem in four decision cycles. The output of the system is in the form of lengthy reports [7, p.183].

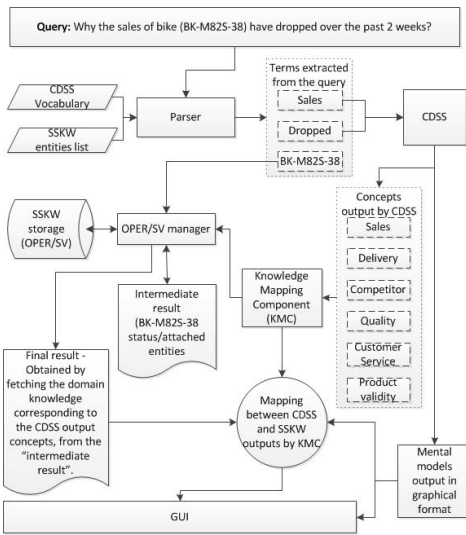


Fig. 7. The decision process of SSKW-based CDSS

SSKW-based CDSS solves this problem as follows (Fig. 7). The system accepts above-mentioned query in natural language, and extracts main terms from it with the help of *CDSS* and *SSKW vocabularies*. The terms found in *CDSS* vocabulary, i.e. *sales*, verb *dropped*, are sent to *CDSS*; while the *SSKW* terms, i.e. *BK-M82S-38* are forwarded to *OPER/SV manager*. *OPER/SV manager* fetches the status of *BK-M82S-38* and related processes, events, relationships, and *SVs*, and keeps them as intermediate result. Meanwhile, *CDSS* extracts all the mental models containing *sales*, with *dropped* as *relationship* (Fig. 8). The *CDSS* then forwards the *list of concepts* from the resultant mental models to the *knowledge mapping component (KMC)*,

which passes them to *OPER/SV manager* while retaining the unique identifiers of these *concepts*. The *OPER/SV manager* searches for these *concepts* in the intermediate result, and sends the result to *KMC*. *KMC* maps *SSKW* and *CDSS* output through the unique identifiers, and passes the output to *GUI*. Thus, when a concept, such as *bad delivery*, is clicked in *CDSS* output (Fig. 8), the corresponding *BK-M82S-38 delivery* report from *SSKW* is displayed (Fig. 9).

6 Comparison between OPER and DW

In *DW*, data is accessed through the dimensions previously declared at design phase, such as *marketing*, *sales*, and *finance*. In *OPER* model however, the business concepts and entities exist individually (but *semantically* associated),

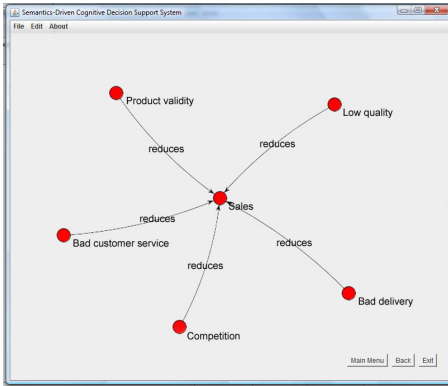


Fig. 8. Output of user query from CDSS

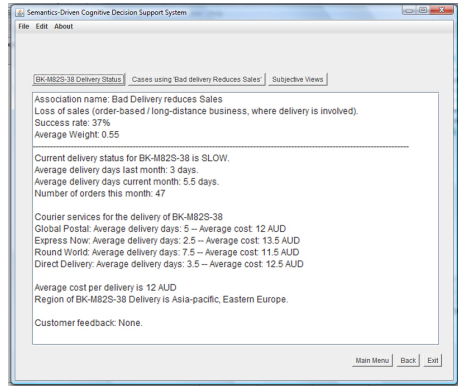


Fig. 9. The delivery report from SSKW

defined as *objects*, *processes* or *events*. Every *object*, *process* and *event*, therefore, becomes the dimension of reference, through which knowledge can be searched and accessed.

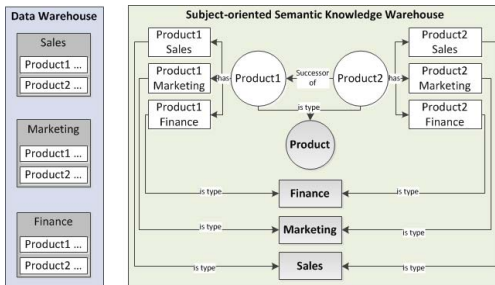


Fig. 10. The difference between the DW and OPER storage

For instance, to get the status of sales for all the products stored, the process *sales* is accessed, which leads to all its child processes, *product1Sales* and *product2Sales*; whereas to output the status of *product1*, the object *product1* and all its processes (*product1Sales*, *product1Marketing*, *product1Finance*) are fetched. This way, OPER enables the access of *relevant* knowledge in efficient and comprehensive manner, improving decision support.

7 Conclusion and Future Work

Considering the significance of knowledge in today's complex business environment, a *subject-oriented semantic knowledge warehouse (SSKW)* has been proposed, which stores knowledge in a unified, meaningful and subject-oriented manner using *subject-oriented* and *semantic* approaches. It facilitates intuitive and efficient communication with CDSS, prevents *loss of knowledge*, improves knowledge precision, and ensures efficient delivery of knowledge.

This can be very powerful tool for decision maker, as the domain knowledge is not divided, but encapsulated through the *object*, *process* and *event* constructs (Fig.10). For instance, to get the status of sales for all the products stored, the process *sales* is accessed, which leads to all its child processes, *product1Sales* and *product2Sales*; whereas to output the status of *product1*, the object *product1* and all its processes (*product1Sales*, *product1Marketing*, *product1Finance*) are fetched. This way, OPER enables the access of *relevant* knowledge in efficient and comprehensive manner, improving decision support.

References

1. Ikujiro, N.: The knowledge-creating company. *Harvard Business Review*, 96–104 (1991)
2. Nemati, H.R., Steiger, D.M., Iyer, L.S., Herschel, R.T.: Knowledge warehouse: An architectural integration of knowledge management, decision support, artificial intelligence and data warehousing. *Decision Support Systems* 33(2), 143–161 (2002)
3. Garcia Perez, A., Mitra, A.: Revisiting knowledge warehousing: theoretical foundations. *Int. J. Bus. Inf. Syst.* 3(6), 572–586 (2008)
4. Dymond, A.: The knowledge warehouse: The next step beyond the data warehouse (2002)
5. Wade, A.E.: Hitting the relational wall (1998)
6. Chen, J.Q., Lee, S.M.: An exploratory cognitive dss for strategic decision making. *Decision Support Systems* 36(2), 147–160 (2003)
7. Niu, L., Lu, J., Zhang, G.: Cognition-Driven Decision Support for Business Intelligence. *SCI*, vol. 238. Springer, Heidelberg (2009)
8. Yadav, S.B., Khazanchi, D.: Subjective understanding in strategic decision making: An information systems perspective. *Decision Support Systems* 8(1), 55–71 (1992)
9. Fu, X., Wei, H.: Cognition inspired object oriented knowledge warehouse architecture. *Journal of Software* 5(9) (2010)
10. Drias, H., Aouichat, A., Boutorh, A.: Towards incremental knowledge warehousing and mining. In: Omatu, S., Paz Santana, J.F., González, S.R., Molina, J.M., Bernardos, A.M., Rodríguez, J.M.C. (eds.) *Distributed Computing and Artificial Intelligence*. AISC, vol. 151, pp. 501–510. Springer, Heidelberg (2012)
11. Goutam Kumar, S.: Business intelligence computing issues. *Ubiquity* 2007, 1 (2007)
12. Pedersen, K.V.: A framework for a clinical reasoning knowledge warehouse. In: *IEEE Proceedings of the IDEAS Workshop on Medical Information Systems: The Digital Hospital*. IEEE (2004)
13. Chabaliér, J., Capponi, C., Quentin, Y., Fichant, G.: Isymod: a knowledge warehouse for the identification, assembly and analysis of bacterial integrated systems. *Bioinformatics* 21(7), 1246–1256 (2005)
14. Kassel, S., Grebenstein, K., Tittmann, C.: A knowledge-based decision-support-system for e-commerce. In: *EUROMEDIA 2005: 11th Annual Euromedia Conference*, Eurosis, Ghent (2005)
15. Zhang, H., Liang, Y.: A knowledge warehouse system for enterprise resource planning systems. *Systems Research and Behavioral Science* 23(2), 169–176 (2006)
16. Sun, X.: Osln: An object-oriented semantic link network language for complex object description and operation. *Future Generation Computer Systems* 26(3), 389–399 (2009)
17. Wade, A.E.: Hitting the relational wall (April 13, 2010, 2005)
18. Levy, M.: A conceptual model of a knowledge warehouse. In: *Enterprise Information Systems Design, Implementation and Management: Organizational Applications*, pp. 148–161. IGI Global (2011)
19. Levy, M., Pliskin, N., Ravid, G.: Knowledge warehouse for decision support in critical business processes: Conceptual modeling and requirements elicitation. In: Schuff, D., Paradice, D., Burstein, F., Power, D.J., Sharda, R. (eds.) *Decision Support*. *Annals of Information Systems*, vol. 14, pp. 131–148. Springer, New York (2011)
20. Fu, X., Wei, H.: Cognition inspired object oriented knowledge warehouse architecture. *Journal of Software* 5(9), 926–933 (2010)

21. Mitra, A., Lau, J.: Challenges of developing an interactive knowledge warehouse within the media industry: Significance of emergent frameworks. In: European Conference on Information Systems (ECIS), pp. 96–104 (2004)
22. Firestone, J.M.: Object oriented data warehousing. Executive Information Systems Inc. (1997)
23. de Abreu, P.F.: Knowledge classes or canonical representation (1996)
24. Sheth, A., Arpinar, B., Kashyap, V.: Relationships at the heart of semantic web: Modeling, discovering, and exploiting complex semantic relationships. In: Nikravesh, M., Azvin, B., Yager, R., Zadeh, L.A. (eds.) Enhancing the Power of the Internet. STUDEFUZZ, vol. 139, pp. 63–94. Springer, Heidelberg (2004)
25. Bush, V.: As we think. *The Atlantic Monthly* 176(1), 101–108 (1945)
26. Zhuge, H.: *The Knowledge Grid*. World Scientific Publishing Co., Inc., River Edge (2004)
27. Harrison, W., Osher, H.: Subject-oriented programming - (a critique of pure objects). *Sigplan Notices* 28(10), 411–428 (1993)