2012

Conceptual Data Vault Model

Vladan Jovanovic  
*Georgia Southern University, vladan@georgiasouthern.edu*

Ivan Bojicic  
*Belgrade University, bojicici@fon.bg.ac.rs*

Follow this and additional works at: [http://aisel.aisnet.org/sais2012](http://aisel.aisnet.org/sais2012)

Recommended Citation  
[http://aisel.aisnet.org/sais2012/22](http://aisel.aisnet.org/sais2012/22)
CONCEPTUAL DATA VAULT MODEL

Vladan Jovanovic  
Georgia Southern University  
vladan@georgiasouthern.edu

Ivan Bojicic  
Belgrade University  
bojicici@fon.bg.ac.rs

ABSTRACT
In this paper a novel conceptual data warehouse (DW) data model, the Conceptual Data Vault (C-DV) Model is presented. The C-DV extends logical and physical Data Vault (DV) Models, to enable conceptual modeling of source data worthy of preserving in a DW staging area as a System of Records. The advantages of DV model, namely flexibility of evolving DW structure using only additions to the model, and high performance of prompt loading, are now extended with a model appropriate during conceptual data discovery and/or analysis phase. Specifically we provided formalization for the C-DV Model as a platform independent conceptual DW model using meta-modeling, as well as defined a patterns based mappings from source data (conceptual entity relationship) models into the C-DV model.

Keywords
Conceptual data warehouse model, data vault model, source data mapping patterns

INTRODUCTION
Nowadays database developers would not attempt DB design without first using some conceptual data model, but the same practice does not generally extend to the DW design. Prominent researchers such as (Jarke, 2010; Goldarelli and Rizzi 2009, Malinowski and Zimanyi, 2009) claim that to be one of the major weaknesses of the first generation DW design methods used in industry and thought in academia (Imhoff, Galemmo and Geiger 2003; Kimball and Ross 2002; Thareja 2009; Watson 2002). The practice evolved a Dimensional Model (DM), visualized with a star like schema, into a de facto industry standard for logical design DW representation (Adamson 2010, Kimball, 2011, Laberge, 2011). On the research/academic side, a number of conceptual models had been proposed. The one with a growing industrial adoption, the Dimensional Fact Model (DFM) (Golfarelli, Maio and Rizzi, 1998), readily translates into the DM see the internationally acclaimed textbook (Golfarelli and Rizzi 2009). Other conceptual DW models, such as StarER (Tryfona, 1999), DWCDM (Franconi, 1999), EventERM (Baekgard, 1999), ME/R (Husemann, 2000), MAC (Tsois, 2001), ODWC (Dori, 2008), GOOMD (Sarkar, 2009) etc. (see: Phipps, 2002; Rizzi, 2008; and Niedrite, 2009 for detail analysis, are also generally compatible with the DM.) Note that in all of the conceptual models listed above there is little or no attention paid to the DW data staging area modeling. The numerous advances in information technology, notably improved price performance of storage, multi-core processors etc. lead to envisioning the next generation of DW. The one clear vision is the new DW 2.0 architecture (Inmon, 2008). In addition, the significance of a permanent staging data area for future enterprise DW is recognized in (Damhof, 2008) and particularly well by (Linstedt, 2011) since the 90’s. The state of the art DW design models and methods are now expected to address requirements for the next generation of DW. The two recent temporal models, grounded in sixth Normal Form (6NF), are the Anchor Model (Romback, 2010), still researching physical implementation consequences, and the DV model (Linstedt, 2011) represent promising ideas within the DW 2.0 and beyond. The DV model is particularly significant for practice due to a proven implementation record and evolved patterns of transformations see for example (Casters 2010). The DV model is now perfecting its logical model, see specification of the 1.0.9 version (Linstedt, 2011). The DV modeling practice continue utilizing DM as the logical DW model of the ‘publishing’ area- reporting to users (Linstedt, 2011) but does not have a defined conceptual model (for the staging area). If we are to sum up the state of the art in conceptual and logical DW modeling, we may be justified in recommending the DFM as an adequate conceptual model for reporting side of DW, the DV Model as preferred logical and physical model for the DW staging area, and the Dimensional Model as the standard logical model for DW reporting area. This leaves research into conceptual model for DW staging area fairly open (the only contender being the Anchor Model). The research in the area of physical models and performance improvements for DW reporting area is outside the scope of this paper but remains wide open within the DW 2.0 (Inmon, 2008) compliant DW design technology space.

Before explaining the specific innovation presented in this paper, namely a formalized C-DV, it is beneficial to review the advantages of the DV model itself (more information on logical data vault modeling can be found in Wikipedia). A number of problems with the first generation of DW including the narrowing of DW update windows are consequences of required complex transformations of inputs going into an enterprise scale DW and Data marts for presentation. Traditional DW
architectures, presented in (Goldfureli, 2008), share most of the following problems: a) complex updates, b) difficulties in source data reconciliation and integration (of substantive viewpoints), c) growing significance of delays, caused by data extracts not ready for loading and upfront reconciliation, d) slow loading, e) brittle transformations (due to integration close to volatile sources and even more so among reporting viewpoints), f) lack of an integrated system of record (data staging is not persistent), g) lack of traceability from sources to archives, and h) complex reconstructions of schemas (evolving content/semantics not really addressed at all). Those challenges are certainly some of the important requirements for the next generation data warehouse modeling technology. The requirements a, b, c, d, and h represent real architectural problems for DW designs based on traditional Entity Relationship Models and/or Dimensional Models. The f and the g are requirements that were not fully appreciated in most of the early DW systems, nor emphasized by design leaders i.e. Kimbal’s or Inmon’s schools of DW architecture and design (Kimball 2002, Kimball 2010), (Imhoff, 2003), etc., and even in the contemporary state of the practice see (Mundy 2011), (Adamson 2010), and (Laberge, 2011). The DV model enables structural changes by addition only, effectively resolving the c, e, and f requirements. It allows loading from the source into a persistent staging area (the data vault) without delays and by this successfully addresses a, e, and h while relegating b to downstream oriented transformations related to the presentation of data for various users. The DV (Linstedt, 2011) include empirically defined set of metadata such as time, data source, and even input error references (rarely seen in traditional logical DW designs). Because of the all aforementioned, the DV model is recognized as the optimal choice for DW2.0 architecture. Given the significant benefits of the DV approach for flexible DW, a development of a formalized conceptual data model compatible with DV captured our attention. The main reasons for explicating C-DV model are: a) to represent and analyze data requirements early, b) to elaborate domain ontology regardless of readily available data sources, and c) to establish formal transformations from source data models to a C-DV and later on from C-DV to logical DV Models. In case of a DV model work is done on transformations of logical and physical DW (staging area) DV models into the logical Dimensional Model and variations of physical models, for example open source Pentaho Kettle (Casters, 2010) and what is left open is a front end elaboration of an appropriate conceptual model. Among the uses of a conceptual model for the DW staging area i.e. a data vault is analysis of true identities (identifiers) and essential compatibility of source data (establishing new links between source systems) before DW design and during major expansions in terms of scope.

CONCEPTUAL DATA VALUT MODEL (C-DV)

The C-DV Model is defined here in a tradition of model driven development using an appropriate platform independent metamodel and a few metaconcepts supporting clear explanation. The metamodel is targeted to express semantics of the complex DW systems and to mediate various transformations. C-DV Metamodel is depicted in Figure 1 using UML notation.
All CDV-Concepts are situated in an implicit real context of Time (typically bi-temporal), a Place (local) for a System-Subject, according to a Convention (standard, law, contract etc.)- reference meta entities to be exposed in C-DV instances. The primary CDV-Concepts, P-Concepts, are ‘defined’ as follows. The P-Concept is a notion of interest whether it is real or abstract (as always only observable concepts make sense for real DW) and it may exist either as a Hub or as a Link. The Hub is intended to represent major identifiable concepts-entities of interest from the real world. It is required that every Hub entity can be denoted by a unique identifier. For a C-DV we represent this identity by a Key, and during logical DW modeling those natural keys are typically complemented (functionally substituted) with surrogate keys. This ‘Key’ property differentiates concrete entity instance from all other instances and may serve as the only (permanent) “truth” about the entity. All other properties including connections to other entities are changeable over time, and according to the essential DV requirement, the only allowable updates will be additions. Link represents relationship among Concepts. Both, Hubs and Links may be involved in such relationships. Dependency is a special form of a Link which may express “has a” relationship in UML (or “weak entity” in various Entity Relationship notations). Another type of Link is the Generalization which represents “is a” relationship. At last, Association is type of Link used to represent all other relationships among Hubs. Furthermore, in the context of C-DV modeling n-ary (i.e. n>2) relationships are allowed, enabling mapping of semantic models not restricted to binary associations (n=2). A Hub or Link context is provided by Satellites. Essentially in a Satellite is used to associate a Hub (or a Link) with (a data model) attribute. A Hub instance has to be related to at least one Satellite instance, which is not mandatory for a Link. On the other hand, every Link is associated with min two Hub instances. In a C-DV Satellite contains one Attribute. This restriction satisfies the 6NF (Date), but is relaxed in Data Vault models (Linstedt, 2011). Concepts may be related to more than one Satellite. The Key may be Composite Key if it is composed of several Values. Any (business) Key instance identifies one and only one Hub. The term Value stands for a datum (singular data value) in a traditional sense. The set of pattern transformations from data model to the C-DV concepts is shown using platform independent UML model (to describe source concepts). The Figure 2 is sufficiently (structurally) rich while simple and recognizable example depicting EU rental car operation, presented using the UML Class Diagram notation.

![Figure 2. EU Rental Car Model](image-url)
Entity and Attribute(s) Pattern: Every entity is converted to a Hub related to Satellite(s) which contain(s) the source entity’s attribute(s):

![Figure 3. Entity and Attribute (s) Pattern](image)

To save space only single Satellites are going to be shown, with an understanding that to maintain the 6NF expectation, each attribute would require a separate satellite; this will assure principle of independent updates by additions only. A logical and physical DV models with ‘inseparable’ multi-attributes will use Satellite for such sets of attributes instead of one for each.

Association Pattern: Associations are transformed to Link entity (many-to-many cardinalities), regardless of the association cardinalities in the source model.

![Figure 4. Association Pattern](image)

Aggregation (Composition) Pattern: Both Classes involved becomes Hubs, and Aggregation (Composition) is transformed to Link concept. All attributes are assigned Satellites (total in this example would be 6 if id and number are respective natural keys of Hubs).

![Figure 5. Aggregation (Composition) Pattern](image)

Association Class Pattern: Classes involved in association are transformed to Hubs while the Link which relates these Hubs, gets Satellite(s) with additional attributes (of the Association Class). Note that a reified Association Class (with a proper identifier) that participates in associations of its own would be handled as a Hub connected to the Link representing the original Association Class.

![Figure 6. Association Class Pattern](image)
Generalization Pattern: Every specialization relation is converted to a corresponding Link:

Figure 7. Generalization Pattern

Recursive Association Pattern: Link related to Hub two times, once for each role.

Figure 8. Recursive Association Pattern

Note that we recommend using Idef1X standard notation (instead of the UML) for efficient reengineering of physical Relational DB models prior to C-DV modeling, due to the Idef1X’s smallest cognitive distance from the Relational Model.

CONCLUSION

What is achieved by explicit definition of a DV compatible conceptual model?

First, a platform independent representation appropriate for conceptual modeling of a DW, seen as a system of record i.e. persistent staging area, is now made available at least to DV practitioners. Second, C-DV is a 6NF compliant model effectively eliminating updating dependencies among attributes. The third accomplishment is conceptual recognition of Time and Convention context metadata categories (in addition to the DV implicit Time and Source-System) providing a lot more sophisticated analysis and traceability (compared with traditional DW models). We intend to expand experimental work and collect evidence for that claim. Furthermore, a systematic algorithm for transforming conceptual source data models into a C-DV is provided in a form of patterns that may serve as clear guidance for DW modelers and DW design tool builders.

The original Data Vault resolved major (flexibility and performance related) DW problems by elevating staging area into a persistent system of records. Providing an explicit C-DV model to facilitate developers working on DW design will allow them to start with a platform independent conceptualization appropriate for requirements and/or architectural analysis as well as for design automation.

How C-DV compares with the Anchor model (we intend to elaborate on this elsewhere)?

In a nutshell, C-DV uses only three concepts i.e. (Hub, Link and Satellite) vs. Anchor model’s four (Romback, 2010). Furthermore the C-DV (see Fig. 1) provides three additional explicit categories for the context i.e. metadata reference types, Anchor model recognize (implicitly) only Time, thus severely limiting its usefulness for desirable system of record DW in industrial multi-source enterprise scale systems which is typically needed in complex environments.

Our work is progressing in extending DW practices to C-DV design integrated into a complete DW design automation and will include transformations of: a) C-DV model into a Logical Data Vault model (L-DV), b) DV constraints and pragmatic workarounds into a relational data model, c) L-DV into logical DW model for presentation (that is the Dimensional Model), d) C-DV to DFM, and f) C-DV to Dimensional Model. Furthermore we are working on defining extensions/revision for standard DW interchange model (Poole 2002) appropriate for the second generation DW architecture (i.e. DW 2.0).

The last question for some may be: is the data vault a DW? A persistent staging area is a (portion of) on enterprise DW.
REFERENCES