Issues in Information Systems
Volume 13, Issue 1, pp. 121-132, 2012

PERSISTENT STAGING AREA MODELS FOR DATA WAREHOUSES

V. Jovanovic, Georgia Southern University, Statesboro, USA , vladan@georgiasouthern.edu
I. Bojicic, Fakultet Organizacioninh Nauka, Beograd, Serbia boicici@fon.bg.ac.rs
C. Knowles, Georgia Southern University, Statesboro, USA, curtis_m_knowles@georgiasouthern.edu
M. Pavlic, Sveuciliste Rijeka/Odjel Informatike, Rijeka, Croatia, Mile.pavlic@ris.hr

ABSTRACT

The paper's scope is conceptual modeling of the Data Warehouse (DW) staging area seen as a permanent system of records of interest to regulatory/auditing compliance. The context is design of enterprise data warehouses compliant with the next generation of DW architecture that is DW 2.0. We address temporal DW aspect and review relevant alternative models i.e. compliant with DW 2.0, namely the Data Vault (DV) and the Anchor Model (AM). The principal novelty is a Conceptual DV Model and its comparison with the Anchor Model.

Keywords: Conceptual Data Vault, Persistent Staging Area, Data Warehouse, Anchor Model, System of Records, Staging Area Design, DW Compliance, Next Generation DW Architecture, DW 2.0

INTRODUCTION

The new DW architecture 2.0 [14] is an attempt to define a standard for the next generation of Data Warehouses (DW). One of the explicit requirements in the DW 2.0 is to support changes over time (while preserving the entire history of data and its structure). A DW 2.0 staging area may be explicitly recognized as a System of Records [15] or an Enterprise DW, see [4], and [20]. It is an area clearly separated from a reporting oriented data area i.e. Data Marts.

This arrangement pushes some complex data transformations closer to the end users but allows loading without delays. Contemporary DW design, (see [1], [13], and [16]), uses Dimensional Model (DM) for the logical modeling of the reporting side of the DW. The Dimensional Fact Model (DFM) [9], [10] is becoming the conceptual DW model leading to the detail design of DM [23]. Alternative models, mostly used experimentally in the past, for example [2], [7], [12], [21], [27], [29], and [30] are also compatible with the DM.

The most recent works are shifting attention to ontology, such as in [11] and [22] starting from existing data sources, provide new focus on empirical data samples [3] in collecting user requirements, extend automatic transformations in MDA style [33] or propose yet another graph modeling visualization [34], but all are still oriented towards reporting side of DW, compatible with Dimensional Modeling. We found no theoretical/conceptual analysis of permanent staging area in the DW literature. It is indicative that even when design of staging area is explicitly acknowledged as emerging research area of increased interest, as in the whole chapter on data staging in [9], all the references cited are dealing with the ETL processes.

These observations strongly motivated focus of this paper onto conceptual modeling for the DW staging area of a special kind, namely a permanent system of records. To clarify the situation let's start with a review of two grounding ideas (requirements) enabling 'total recall' of both content and the structure for rapidly evolving DWs. Figure 1 informally contrasts a real system (top line) with corresponding OLTP Database (DB), shown in the middle line, and the DW (both its staging and the reporting areas) shown in the bottom line. For a permanent staging area the first problem is preservation of the entire history of data (content) without redundancy, and the second problem is how to assure that all structural changes are similarly preserved.

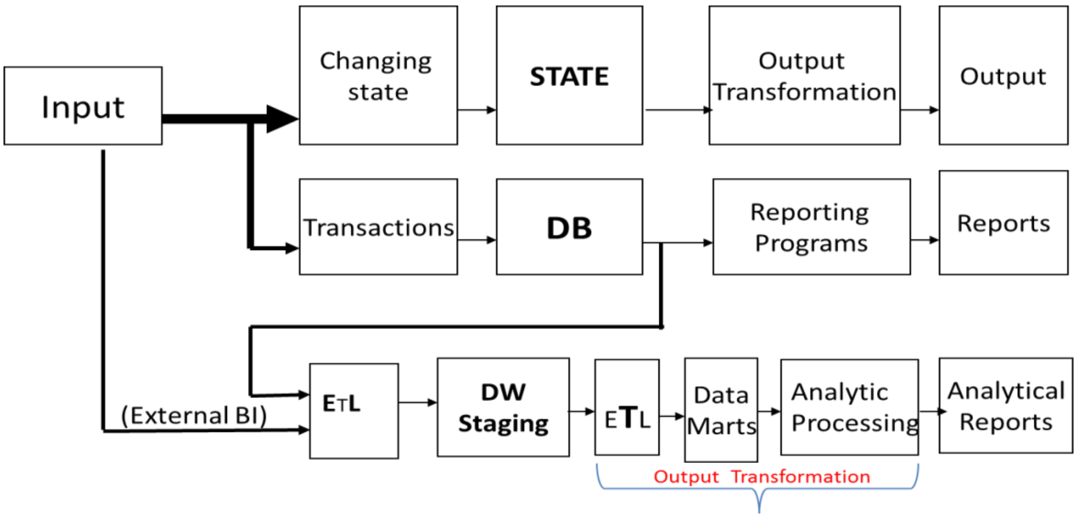


Figure 1. Scope of DW Staging vs. DW Reporting Area

Note a recurrence of a state based system model pattern in Figure 1: first, a real system state (abstractly understood as minimal information about the system) is changed by inputs (events influencing the system) but observed trough some output transformation. Second, the same information about inputs is (via transactions) updating operational on line transactional DB, and system information i.e. state is observed via reporting programs. Finally, DW Staging (as complete information) is cumulated and integrated with external information so that new, DW specific, output transformations can publish any demanded information to end users. This separation of reversible i.e. light transformations (before loading permanent staging area as the system of record) and more complex (mostly rule based) and frequently irreversible i.e. heavy transformations (downstream towards reporting) is the essential idea advocated here as well as by Dan Linstedt in [18], [19] and [20]. To fully explain the implication of complete information, in a sense of historical information about the past and present, and subsequently ground any DW system design, we need to address temporal aspect of data (changing values in time without anomalies, by analogy with traditional normal forms).

Temporal Aspect- Data Changes and the new Sixth Normal Form

Let us address the temporal aspect and clarify ideas per [5] and [28] enabling a conceptual solution of the first problem (preserving all data values without redundancy). The goal is a comprehensive DW staging area able to store not only current data, but also past data without any alterations or loss. At the time point where change occurs, a new snapshot can be created and delimited in **time** using *since* and *to* (moments) forming a valid time interval. Capturing snapshots generates a historical record of data. Date [5] established that records must be organized to decompose data into a series of irreducible sixth normal form (6NF) compliant tables; formally "A relvar (table) *R* is in 6NF if and only if *R* satisfies no nontrivial join dependencies at all, in which case *R* is said to be irreducible." Since the paper [28] it is well established that both transaction time and valid time are needed to preserve information of state as it was known at the time as well as what was true at the time (both from the vintage point of now); this is the key requirement of bi-temporal (two orthogonal time referencing concepts) models to be explored in this paper. Figure 2 illustrates relation (SUPPLIER) in the fifth NF and its temporalized variant that replicates all unchanged data each time something gets updated.

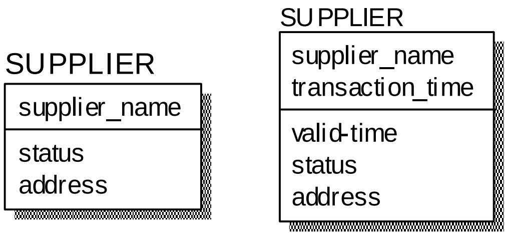


Figure 2. Non 6NF approach

The Figure 3 shows decomposition into a set of 6NF projections, allowing for independent changes of all attributes (and, with the use of a surrogate key, for independent changes to the natural key as well- obviously this is of very significant practical importance for system integration).

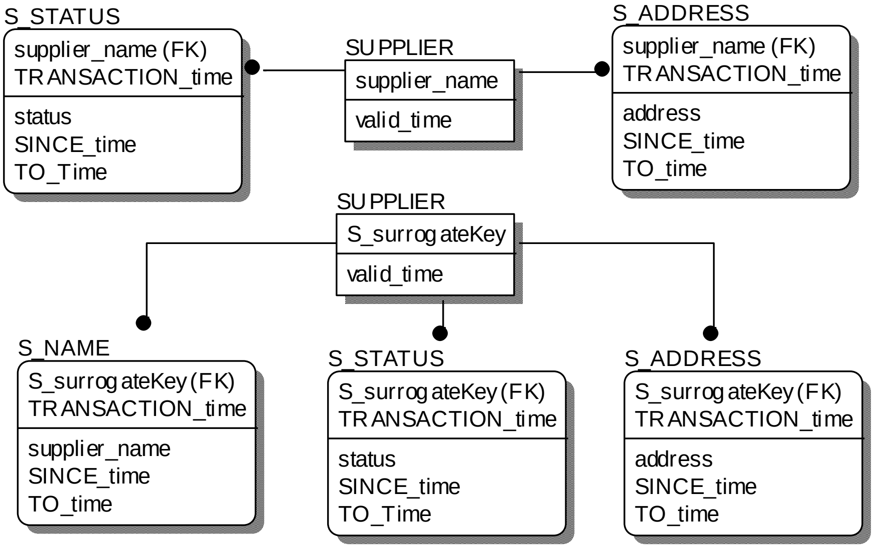


Figure 3. Temporal 6NF Schemas

Understanding a solution for the key problem- preserving structural changes (in addition to attribute/data value changes) requires elaboration of conceptual modeling approaches and models that address this requirement explicitly.

CONCEPTUAL DATA VAULT MODEL

Let's start with a model of some source database, shown in Figure 4, and use it to illustrate two alternative conceptual models for a Data Staging area, both temporalizing relationships in addition to temporalizing entities and attributes. For reengineering physical DB data source models, prior to C-DV modeling, the Idef1X notation is recommend, to maintain minimal cognitive distance from (source databases and) the relational model. Any ER notation, as well as UML etc. may be used and use of Idef1X here (with an important restriction of not directly allowing any M:M relationships) only simplifies exposition as a visualization model that is very directly mappable to the realtional model (every entity becomes relatioin and every relationship becomes FK).

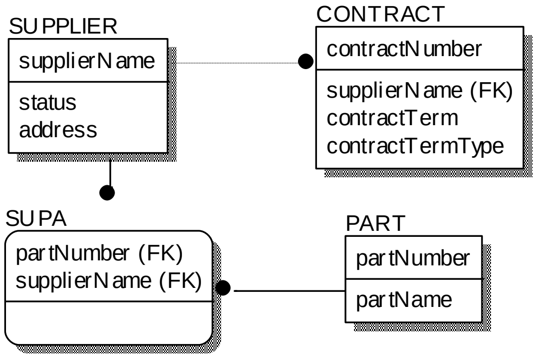


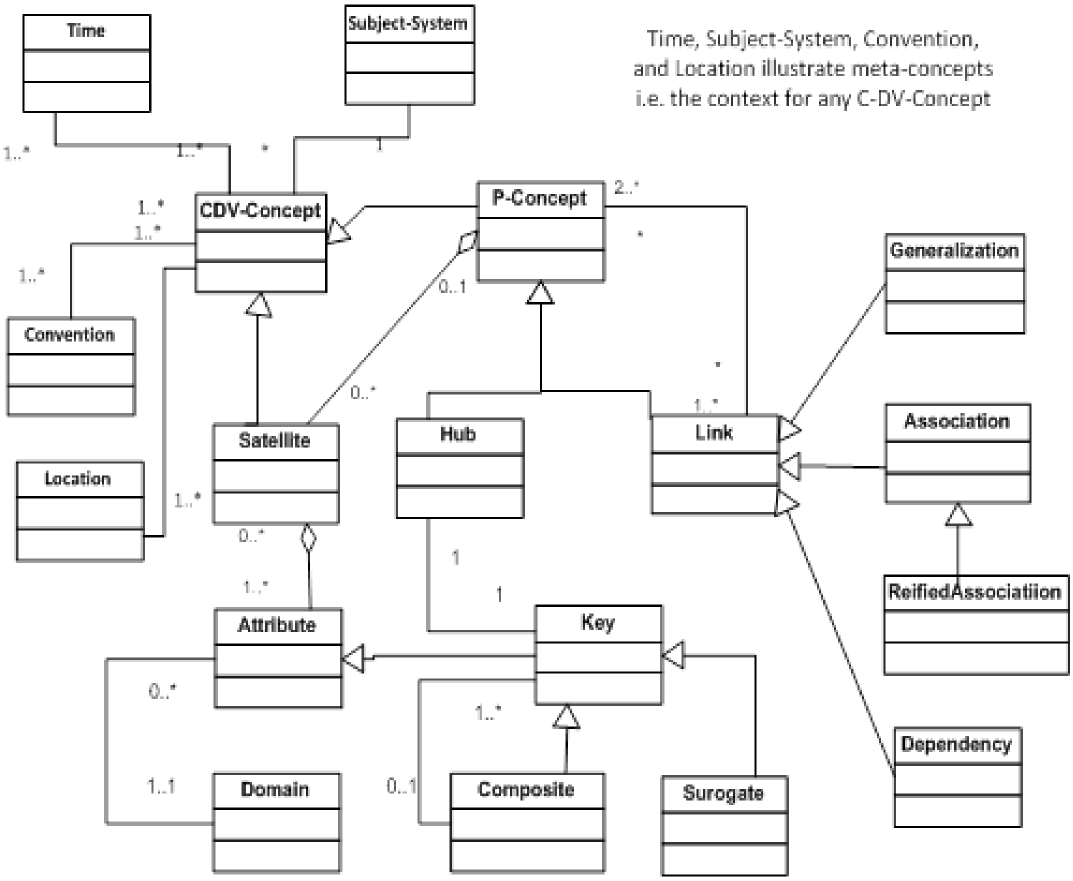
Figure 4. A conceptual model of illustrative source DB

The Data Vault model, developed in industry by Dan Linstedt, (see [18], [19], [20], and Wikipedia article [32]), is primarily focused on logical and physical design of a DW staging area as a system of records and an Enterprise DW. The DV is not used for modeling Data Marts. In [17] Knowles demonstrated suitability of both the DV and the Anchor Model for a DW 2.0 [14] compliant DW design. The C-DV model is an essential representation suitable for analysis of source databases and modeling of integrated DW. The C-DV is our abstraction of the traditional (logical and physical)

DV model. The only concepts used by C-DV are Hubs, Links and Satellites. In [15] the C-DV is defined via a metamodel presented in UML. The Figure 5 presents a slightly refined C-DV metamodel. This paper focuses on and demonstrates the use of the C-DV, in comparison with the use of the Anchor Model, for modeling permanent DW staging area.
All of C-DV constructs i.e. the **CDV-Concepts** (from Figure 5) are situated in an implicit real context of Time (typically bi-temporal) and Location (of interest in precise data sourcing) for a System-Subject (application producing data to be used as the source), and eventually according to some reference Conventions (standards, laws, contracts etc.). All of these meta-entities are references present in C-DV instances as implied metadata (and will not clutter any of the C-DV diagrams to be shown here). The primary C-DV concepts, the P-Concepts in the figure, are 'defined' as follows. The **P-Concept** is a notion of interest whether it is real or abstract (as always only the observable concepts make sense for real DW, so please note substantive tie with the ontology, literally of what exists) and it may exist either as a Hub or as a Link. Note a unique C-DV capability of a Link to connect to a Link, not just to a Hub (as restricted per DV), see Figure 5. This is one important difference between the C-DV and the DV models, namely DV models frequently use Reference Tables as a form of 'static' Hubs encapsulating their Satellites into traditional non-historicized relations, while our C-DV implicitly historicizes everything.

To visualize C-DV any Entity Relationship Atribute notation can do but using formally standardized i.e. weel defined notations such as the Idef1X, or the UML is recommended. In the C-DV representation, Figure 6, all metadata are abstracted (from Figure 5) for simplicity. Some of the metadata defined in [15] are: source system, location of data entry, transaction time of update (load), and the valid time interval (pair). The industrial practice reported in [19], and [20] recommends using color to distingush essentail types; for example light blue for Hubs, red for Links (in this paper we use a very light bieg (so that shade of color remains visible in all black and white reproductions), and light yellow for Satellites.

A **Hub** is intended to represent entities of interest from the real world. It is required that every Hub entity can be denoted by a unique identifier, at the C-DV level we represent it by a Key, and during logical DW modeling those are typically replaced (complemented) 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 or connections to other entities are changeable over time.



Time, Subject-System, Convention, and Location illustrate meta-concepts i.e. the context for any C-DV-Concept

Figure 5. Conceptual Data Vault Metamodel in UML

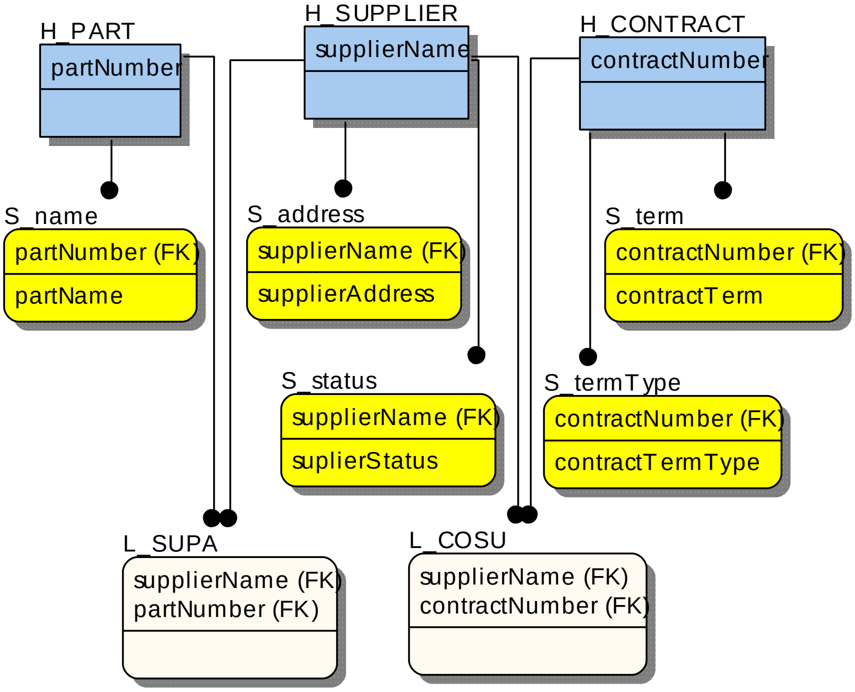


Figure 6. Conceptual Data Vault (C-DV) temporalizes all Links and Satelites

A **Link** represents a relationship among Concepts. Both, Hubs and Links may be involved in such relationships (the DV model [20] did not specify Link to Link connections). A typical data relationship is a Link representing binary or n-ary relations/associations among Hubs. Furthermore, in the context of the C-DV use, modeling of n-ary relationships is allowed. This show a potential to broaden scope of C-DV mapping from semantic data models not restricted to binary associations. A Hub or Link context is provided by **Satellites**. Essentially a Satellite is used to associate a Hub (or a Link) with (a data model) attribute. In the C-DV a Satellite contains one Attribute. This restriction satisfies the 6NF [5], but is relaxed in all implemented Data Vault models [20]. Hub and Link Concepts may be related to more than one Satellite. DV metadata, not shown here for simplicity, are the two time references, error status, and source data system for all instances in a data vault. The data vault name itself comes from the fact that no deletes and no alteration of any kind are allowed in a system of record (supporting the audit and strict compliance requirements). This in particular is making the Data Vault model very interesting for high security and safety environments including financial, government and large science-data data warehouses. As the C-DV is for conceptual modeling, primarily of available source, data some may question how it supports recursive, dependency and generalization relationships among other? That is illustrated on Figure 8, but for the full set of basic C-DV mapping patterns see [15] where the UML into C-DV model transformation is defined.

How does DV resolve the second problem? It is easy to explain some typical but smaller changes first, say a 1 to M relationship became M to M. Such changes cause no problems as all model relationships are already represented by the (essentially M to M) Links within a DV (and within the C-DV). How about more complex changes, for example merging several DB/DW sources? The key new things (for integration) will be new Links relating existing Hubs from separate DWs, and possibly new Satellites (i.e. attributes) at the created 'top' level as well as genuinely new Links whenever new information became of interest. Some of the structural changes are illustrated in Figure 7. It is clear that all the schema/model changes other than additions (such as Link becoming a Hub) can only be modeled at the meta-level in a catalog, and to remain consistent we advocate the DV catalog to be modeled using C-DV (assuring feasibility for implementation is subject of ongoing work and will not be elaborated here due to space limitations).

With the C-DV the burden of data reconciliation for reporting side of DW remains but is relegated downstream from persistent staging area into the presentation area that is towards Data Marts. Consequently most of the analysis and design ideas from traditional DW analysis models such as the DFM [9] and the Dimensional Model see for example [1] and [16], remain highly relevant in designing Data Marts.

The main appeal of DV is that data will be preserved (as in a literal vault) including erroneous ones, and that is a common and important requirement in defense, intelligence, police, banking, and similarly sensitive systems [20]. Furthermore Hubs and their Satellites can be easily isolated via Links and (different) encryptions used within the public and private portions of the system, with obvious implications for enhanced security and/or privacy. A formalized definition of the C-DV, in terms of set theoretic concepts, is provided in Table I.

TABLE I. FORMALIZED DEFINITION OF THE C-DV

Identities	Let *U* be a finite but arbitrarily large set of symbols, which are used as identities.
Data type	Let *D* be a data type. The domain of *D* is a set of data values.
Hub	A Hub H is a string. An extension of a hub is a subset of *U.*
Time type	Let *T* be a time type. The domain of *T* is a set of time values.
Hub Role	A Hub role *R* is a string. Every Hub role has a type, which is a Hub.
Link	A link L is a set of at least two Hub (or Link) roles and a time type *T*. An instance *Li* of a L= {R_1,…,R_n, *T*} is a set of pairs [R_i, v_i], i = 1,…,n and a time point *p*, where R_i is one Hub/Link role, v_i Є *U*, *p* Є *T*, and $n \geq 2$. An extension of a link L is a set of instances of *Li*.
Link Role	A Link role *R* is a string. Every Link role has a type, which is a Link
Satellite	A satellite *S* is a string. An S has a Hub or a Link for a domain, a data type *D* for range, and a time type *T* as time range. An extension of a Satellite is a relation over *I x D x T*.
Identifier	Let *L* be a Link. An identifier for *T* is a subset of *T* containing at least one Hub/Link role, and every identifier for *T* must contain *T*
C-DV	{U x H x L x S x T x D} as per Metamodel

Two small structural changes, illustrating some plausible scope extensions of the running example of a DW, are shown, in Figure 7 (remember there are never any deletions from a C-DV model). Note a Link to Link connection illustrated with L_PASUDE (except for the Links we discourage use of abbreviations). This is a conceptual link, in fact structural changes are at the level of metadata, reflected in the catalog (meta model), which for any real data vault implementation we recommend also be modeled and built as a data vault to preserve entire history of changes (research on this is ongoing).

Conceptualizing data using data vault C-DV data model is new and some may ask how, in principle, situations traditionally modeled with advanced associations (for example in Idef1X) are to be handle with C-DV . The Figure 8 shows some such patterns.

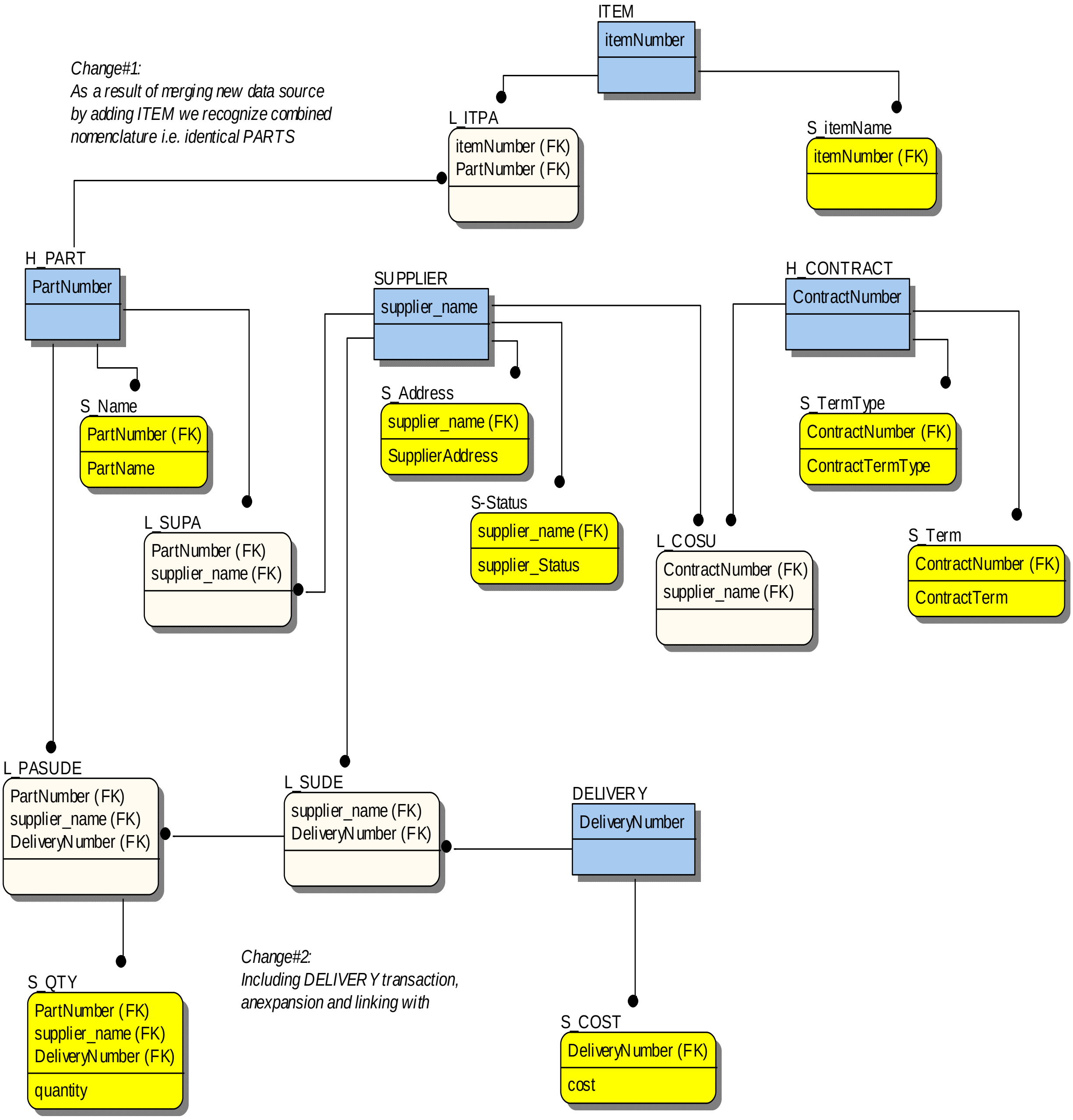


Figure 7. Typical structural changes to a C-DV are visible as model expansions

The Figure 8 shows C-DV equivalent representations for some typical data modeling segments and can serve as a guide in conceptual modeling or during an analysis of existing data models.

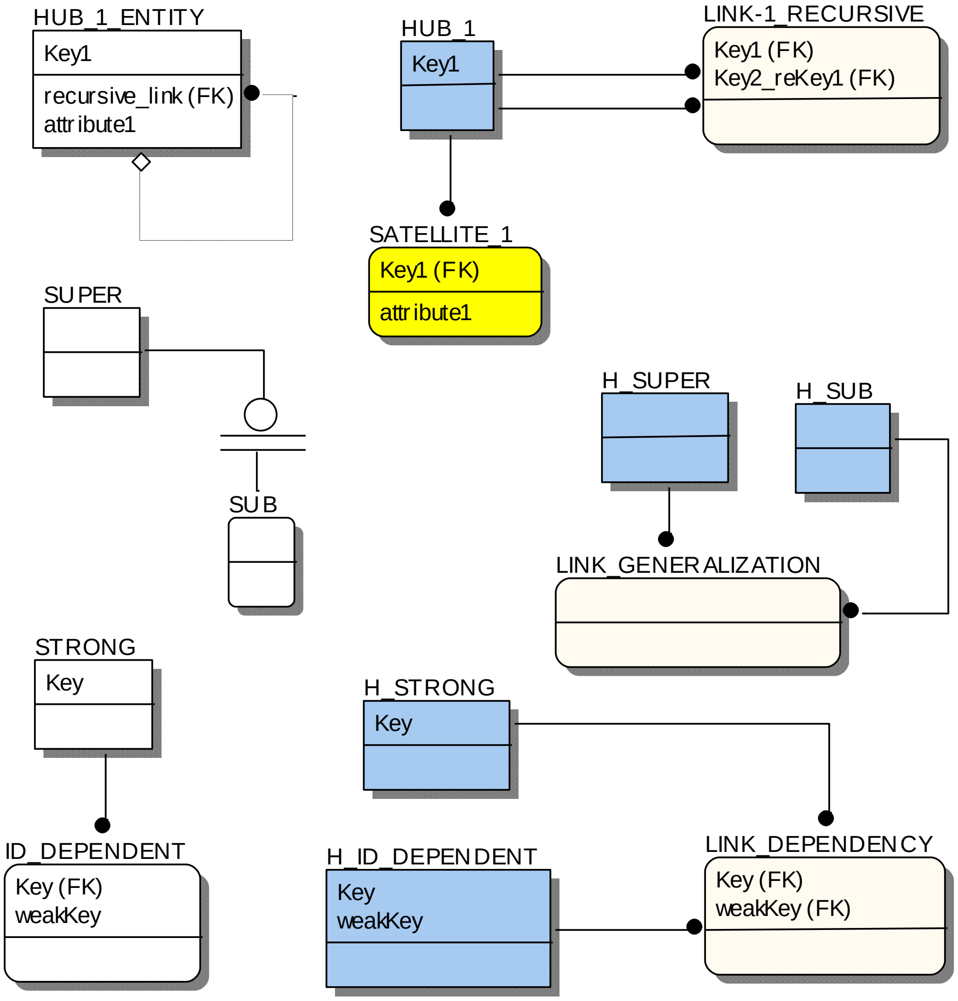


Figure 8. C-DV representation of some semantically advanced constructs

What is the expected gain from a Conceptual Data Vault approach, naturally combined with DV for logical and physical DW modeling?

Figure 1 indicated some separation of Loading from output bound Transformations, i.e. loading becomes immediately enabled (no need for batch update windows and collating source data) and with minimum processing (typically a parallel real time capture [20] and similarly [24]). On the Data Mart i.e. publishing side, a lot more of 'compare and contrast' as well as other investigative opportunities are now preserved. Apparently this would require extensive transformations as needed by (and when needed for) end users. As reported in the [24], 'divorcing' most of the traditional ETL 'reconciliation' complexity from the data loading side of a DW is resulting in significant performance increases in practice. Furthermore, due to a generic structure of DV, its consistent use of only three constructs, efficient pattern based automation is feasible (making a preparation of Data Marts in DV approach competitive with traditional approaches). More research is needed to confirm this and similar claims made in [26] and informally on the blogosphere.

ANCHOR MODEL

We will compare our C-DV model with the standard Anchor Model here using the same source example from Figure 4 for side by side illustration. The Anchor Modeling defined in [24], [26], and [25] provides a graphical notation for conceptual database modeling similar to the Entity-Relationship (ER) modeling, with extensions for temporal data. The Anchor Model is using four constructs: the Anchor, the Attribute, the Tie, and the Knot *(*see illustration example from Figure 4 specified by Table II and shown as corresponding Figure 9). It also uses color coding to visually separate Anchors (Red) and Ties (Gray). Notice an inconvenience; one can't simply use a style in a basic notation (like Idef1X was directly useful for C-DV) as double lines are selectively applied in addition to coloring schema (red for Anchors, gray for Ties, and plain for Knots). This is somewhat compensated by the availability of an open source tool, (see: www.anchormodeling.com/modeler). One can also observe extensive use of abbreviations in practice of Anchor modeling and thus the need for a legend, see Table II (with the four independent sections, one for each of the primary AM concepts).

The **Anchors** model entities and events, the **Attributes** model properties of Anchors, the **Ties** model relationships between Anchors, and the **Knots** model shared (reference) properties (shown as small rounded squares, see ST and TT on Figure 9). By using a double line on an Attribute or a Tie, a history of changes for the information they model is mandated. In Figure 7, the double line around attributes SUADR and COTRM and the Tie SUPA indicate that a history of changes is kept for supplier addresses, contract terms, and supplier-part relationships. All the pieces of data are tied to moments (points) in time or intervals of time. The valid time moments are modeled as attributes (i.e., the time a part is shipped), and intervals of time are modeled as a historization of attributes or ties (i.e., times during which a part was in stock). Transaction time elements, such as the time information entered or was updated in the database, are handled by metadata and are not included in the anchor modeling constructs at the conceptual level [17].

The source data example (from Figure 4) mapped to DW staging area using the Anchor model (partially historicized to illustrate the Anchor model options and notation) is shown in Figure 9, and Table I is a legend for that model. Such legend seems necessary as the practice of Anchor modeling (and the tool mentioned above) extensively use abbreviations.

Note also that Anchor model developers [24], [25], and [26] consider it a conceptual data model similar to the Entity Relationship model, intended for modeling data bases and data warehouses as well. On the other hand DV model inventor is seeing the DV model as a physical and logical model for DW design.

TABLE II. ANCHOR EXAMPLE LEGEND

ANCHOR	ATTRIBUTE
PA-part	PANAM-PartName
SU-supplier	PANUM-PartNumber
CO-contract	SUNAM- SupplierName
	SUADR- SupplierAddress
	CONUM-ContractNumber
	COTRM-ContractTerm
TIE	**KNOT**
SUPA-SupplierPart	ST-SupplierStatus
COSU-ContractSupplier	TT-ContractTermType

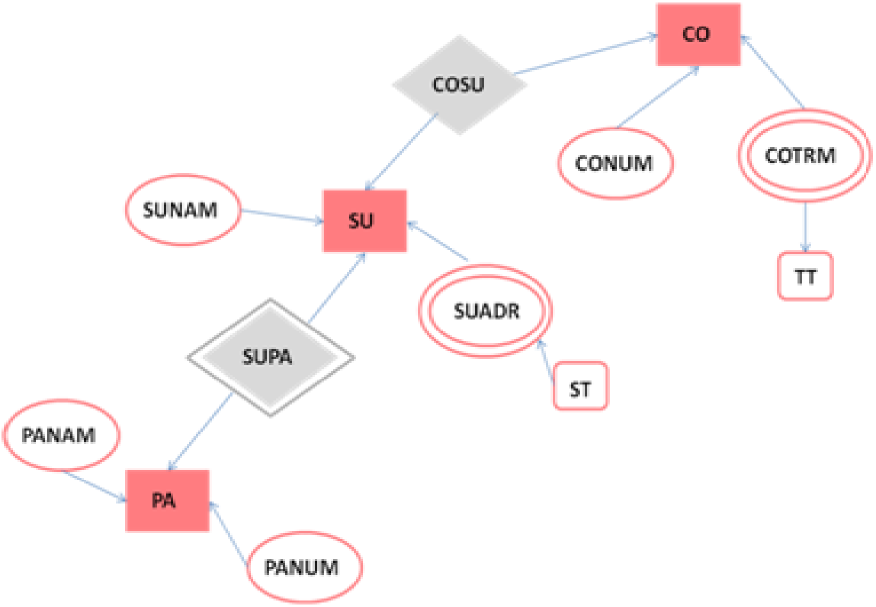


Figure 9. Anchor Model temporalize only double lined

We consider both the C-DV and the Anchor Model adequate for modeling the permanent staging area under the DW 2.0 architecture [14]. The DV model [15], [19], and [20] include data source as metadata, enabling it to carry out the seamless system integration envisioned in the [31].
The difference between C-DV and Anchor Model can be narrowed down by adding metadata (and by analogy with the C-DV) nevertheless Anchor Model's has some minor disadvantages, compared to the C-DV, namely:

a) Premature specification of which of the TIEs (relationships) and/or ATTRIBUTEs are to be historicized and which are not to be,
b) Using separate (double line) icons to express selected historization decisions visually, which represents a cognitive burden (and effectively eliminate direct pragmatic use of existing case tools and/or notations in a manner illustrated here by introducing a simple DV style, as per Figure 6), at the conceptual modeling level,
c) Furthermore, using an additional concept, the KNOT for permanent shared reference types (again prematurely) complicates the matters at the conceptual level, and at this point
d) Lack of extensive industrial implementation work (compared to the DV [19]).

CONCLUSION

The key contribution of this paper is a novel Conceptual Data Vault (C-DV) Model. The C-DV is aimed at extending Data Vault approach [19] to the conceptual modeling for DW permanent staging area and/or enterprise DW as a flexible highly extensible system of records i.e. data vault. Both the DV [19], [20], and the Anchor model sources including the recent paper [24] also point to high performance potential for a DW built upon them. The original Data Vault resolved major (flexibility and performance related) DW problems by elevating transient staging area into a persistent system of records and establishing the resulting data vault as the essential DW component compliant with the new DW 2.0 architecture [14]. The value of the proposed C-DV model is in providing an explicit conceptual model to facilitate developers working on DW design (primarily in the context of the Data Vault approach) that fits requirements for persistent, conceptually integrated data store from different sources, but not necessarily a fully reconciled and cleansed traditional enterprise data warehouse. Reconciliation to the 'single version of truth' standard in traditional Inmon's architecture prior to the DW 2.0 involved (output transformations per Figure 1, that is) transformations that are frequently irreversible and thus better relegated downstream towards user centered and typically somewhat aggregated data sets for reporting known as data marts. This implies three tier system materialization (source, data vault, and data mart) in implementation. With the C-DV model one can start with a platform independent conceptualization appropriate for source data analysis, conceptualizing ontology of existing data requirements and/or early architectural analysis of a data warehouse, as well as for design automation. The extension to metadata (primarily via notions of Location and

Conventions) can support richer and more open models for distributed DW. The fact that C-DV Model can be used to model available data sources, directly leading to logical designs for persistent staging areas by design (ontological integration of Hubs), are making it the only operational model now available for designing DW as systems of records and thus fully compliant with regulations requiring full recall of data (no deletes or overwrites), and provenance analysis in auditing.

The future work is progressing in two major directions, first toward formalization and automation of data vault design methodology, covering logical and physical design of data vault types DW as well as on formal transformations (not limited to derivations of data marts and BI data store systems). The second direction is based on a conjecture that conceptual extensibility with complete history of changes (of structure, i.e. metadata instances, and of the data content) is a separate aspect, and warrant future investigation including a potential of the C-DV model to serve as a conceptual data model for some of the next generation of databases and emerging data stores (including in memory DB, NoSQL DB, BigTable, and in particular the Column-based DB/DW intuitively a 6NF projections). Similarly promising directions are investigating patterns of transformations leading to implementation in the next generation DB/DW/BigData systems, and addressing truly BigData management performance issues for (logically) distributed DW that may eventually exceed zettabytes in foreseeable future.

REFERENCES

[1] C. Adamson, Star Schema – The Complete Reference, McGraw Hill, 2010

[2] L. Baekgard, Event-Entity-Relationship Modeling in Data Warehouse Environments, Proceedings of the ACM DOLAP99 Workshop, Missouri, 1999

[3] L.Corr, Agile Data Warehouse Design, Decision Press 2011,

[4] R. Damhof, The next generation EDW, posted August 2008 at: http://www.prudenza.nl,

[5] C.J., Date, H Darwen,. and N. Lorentzos, Temporal Data and the Relational Model, Morgan Kaufmann, 2002

[6] D. Dori,R. Feldman, and A, Sturm, From conceptual modeals to schemata: An object-process-based data warehouse construction model, Information Systems 2008,

[7] E. Franconi, and U. Sattler, A Data Warehouse Conceptual Data Model for Multidimensional Aggregation, Intl. Workshop on Design and Management of Data Warehouses, Heidelberg, 1999

[8] A. Giordano, Data Integration Blueprint and Modeling- techniques for a Scalable and Sustainable architecture, IBM Press 2011,

[9] M. Golfarelli and S. Rizzi, Data Warehouse Design, McGraw Hill 2009,

[10] M. Golfarelli, D. Maio, and S. Rizzi, The Dimensional Fact Model: A Conceptual Model for Data Warehouses, *International Journal of Cooperative Information Systems*, Vol.7, Nos. 2 and 3, 1998, pp. 215-247,

[11] L He., Y. Chen, N. Meng, and L. Liu, An Ontology-based Conceptual Modeling Method for Data Warehouse, Sciences International Conference of Information Technology, Computer Engineering and Management Sciences, 2011

[12] B. Husemann, J. Lechtenborger, and G. Vossen, Conceptual Data Warehouse Design, Proc of the Intl Workshop on Design and Management of Data Warehouses 2000, Stockholm, Sweden,

[13] C. Imhoff, N. Galemmo, and J. Geiger, Mastering Data Warehouse Design, John Wiley, 2003

[14] W. Inmon, D. Strauss, and G. Neushloss, DW 2.0-The Architecture for the Next Generation of Data Warehousing, Morgan Kaufman, 2008,

[15] V.Jovanovic, I. Boicic, Conceptual Data Vault Model, SAIS Conference 2012 Atlanta, USA, pp 131-136,

[16] R. Kimball, and M. Ross, The Data Warehouse Toolkit, 2ed. John Wiley, 2002

[17] C. Knowles, 6NF Conceptual Models and Data Warehousing 2.0, SAIS Conference 2012, Atlanta, USA, pp160-165,

[18] D. Linstedt, DV specification 1.2 at http://danlinstedt.com/datavaultcat/standards/dqata-vault-loading

[19] D. Linstedt, Super Charge Your Data Warehouse, on Amazon.com, 2011

[20] D. Linstedt, K. Graziano, H. Hulgeren, New Business Supermodel, Lulu 2008,

[21] P. McBrien, A. Selveit, and B. Wangler, An Entity-Relationship Model Extended to Describe Historical Information, CISMOD 92 BNAGALORE, India, 1992, pp. 244-260,

[22] M. Nazri, S. Noah, and Z. Hamid, Using lexical ontology for semi-automatic logical data warehouse design, 5th international conference on Rough set and knowledge technology, Vol. 6401 Springer 2010, p. 257-264 2010,

[23] S. Rizzi, Conceptual Modeling Solutions for the Data Warehouse, in J. Wang, editor, Data Warehousing and Mining, IGI Global, 2008, pp. 208-227,

[24] L. Rönnbäck, O. Regardt, M. Bergholtz, P. Johannesson, and P. Wohed, Anchor Modeling - Agile Information Modeling in Evolving Data Environments, *Data & Knowledge Engineering*, 2010, pp 1229–1253.

[25] L. Rönnbäck, Anchor Modeling – A Technique for Information Under Evolution, GSE Nordic Conference, Stockholm, Sweden, 2001,

[26] O. Regardt, L. Rönnbäck, et all, Anchor Modeling, Proceeding of the 28th Intl. Conf. on Conceptual Modeling. ER Brazil, 2009,

[27] A. Sarkar, Conceptual Design of Object Oriented Data Warehouse: Graph Semantic Based Model, *INFOCOMP Journal of Computer Science*, 2009,

[28] R. Snodgrass, and I. Ahn, Temporal Databases, *IEEE Computer 19(9)*, 1986, pp 35-42,

[29] Tryfona, N. et all, starER: A Conceptual Model for Data Warehouse Design, Proceedings of the ACM DOLAP99 Workshop, Missouri, 1999,

[30] A. Tsois, N. Karayanis, and T. Sellis, MAC: Conceptual Data Modeling for OLAP, Proceedings of the International Workshop on DMDW, Interlaken, Sweden, 2001

[31] P. Wisse, Metapattern- Context and Time in Information Models, Addison Wesley, 2001,

[32] Wikipedia, Data Vault Modeling, last visited 3.29.2012,

[33] L. Zepeda, E. Ceceña, et all., A MDA Tool for Data Warehouse, International Conference on Computational Science and Its Applications, 2010

[34] M Zekri., I. Marsit, and A. Adellatif: A new data warehouse approach using graph, Eighth IEEE International Conference on e-Business Engineering, 2011.