Comparing Anchor Modeling with Data Vault Modeling

Lars Rönnbäck & Hans Hultgren

SUMMER 2013
INTRODUCTION AND BACKGROUND

Modeling the Data Warehouse.

Data modeling techniques for the data warehouse differ from the modeling techniques used for operational systems and for data marts. This is due to the unique set of requirements, variables and constraints related to the modern data warehouse layer.

Some of these include the need for an integrated, non-volatile, time-variant, subject oriented, auditable, agile, and complete store of data.

To address these needs several new modeling approaches have been introduced within the DWBI industry. Among these are Data Vault modeling and Anchor modeling.

Both of these approaches can be classified as forms of Ensemble Modeling.

Ensemble Modeling.

In looking at the several data warehouse modeling approaches that have emerged over the past decade, we can see that there exist a common set of characteristics and features. These include the separation of static and dynamic data, the splitting out of relationships from context, the focus on a central business concept, and the creation of table constellations centered on a unique instance of a concept or domain.

Forms consistent with these characteristics can be said to belong to the family of Ensemble Modeling techniques. Ensemble Modeling is primarily defined by Unified Decomposition – the breaking out of concepts into component parts each maintaining a relationship to the unique identifier representing a single instance of the concept itself.

Effectively Ensemble Modeling embodies Data Warehouse modeling.

Data Vault Modeling.

Data Vault modeling addresses the demands of the data warehouse layer by separating keys (hubs) from context (satellites) from relationships (links). The Hub is based on the natural business key of a core business concept (core entity or domain). Satellites contain logical groupings of context attributes that share common characteristics (such as rate of change, type of data, or source system). Links are used to model the relationships between concepts. Links are not concerned with cardinality but are designed to match the natural unit of work (the relationships between and among relationships).

Anchor Modeling.

Anchor modeling focuses on information changes both in structure and content. It gains its flexibility and temporal capabilities through separating identities (anchors) from context (attributes) from relationships (ties) from finite value domains (knots). Natural keys are not a part of the model itself and composed only as different logical views of each anchor. Attributes are not grouped and may be static or historized to capture changes. Ties model relationships, always have cardinality, and may be historized. Knots are used to constrain a small set of values used to describe states of entities or relationships. A model as a whole may be uni-, bi-, or concurrent-temporal.

Background.

During a BI Podium modeling conference in Summer 2013 both Data Vault and Anchor were presented and also compared. At the end of this conference a form of side-by-side comparison was introduced. This document is based on that comparison and includes a complete discussion on the points presented.
This list above is a copy of the one presented at the summer 2013 modeling conference in the Netherlands. This comparison chart was created by Lars and served as a starting point for the broader analysis.
THE COMPARISON

Comparison Overview.
This list was put together by Lars including also his view of these differences. A couple of additional categories were later added. The following section includes Hans comments based on his view of these compared features.

Impressions and Comments.
Most of the comparison is in line with Data Vault Ensemble with a handful of exceptions. These exceptions have more to do with the theories on how data vault is applied and less to do with structural features of the modeling approach.

• Paradigm.
That data vault is more data driven than domain driven is a misnomer in the industry today. Actually the CDVDM courses teach modeling of the Core Business Concepts which are business defined central concepts (domains, core entities, etc.). The false impression stems perhaps from the history of automation efforts (primarily source system oriented) and the strong focus on auditability. Automation tools should recognize that we are not trying to build a vaulted copy of a source model but rather an integrated, business driven, central model. Therefore pointing automation tools to a central model (information model, logical model) would be more in line with data vault. The auditability is in fact a major capability of data vault models and so maintaining traceability is key. This is the reason for the existence of both Raw and BDW layers in the data vault.

• Grouping.
Anchor has in effect no grouping as attributes each have their own Attribute table. With Data Vault this could in theory also be true (single attribute Satellites).

However the focus of data vault is to have the data architect/modeler design the Satellites based on logical groupings as discussed in the Data Vault section above.

• Stringency.
It is true that data vault does allow for more flexibility in terms of naming conventions and of course also in satellite design. Still there are core components of the modeling pattern itself that are highly structured and defined.

• Schema Evolution.
Certainly there is a range from destructive to non-destructive. On one end of the spectrum you find 3NF close to Dimensional modeling. On the other end you find Ensemble modeling forms which are inherently designed to be agile (highly adaptive to change). Data Vault is perhaps 80% of the way to full adaptability while Anchor would be 99% there. Both forms do still require ETL and DB changes. Please note that generic, NVP and doc-style forms would absorb changes more fully but they are schema-free methods.

• Tightening.
Inherently the Data Vault approach is based on the idea of the insert only EDW. To handle date math it is however sometimes common to update effective end dates in satellites. Note that in uni-temporal Anchor erroneous data is also ‘updated’ (deleted/inserted), but can be handled through concurrent-temporal as insert only.

• Adapting to Change.
All Ensemble forms share a goal of adapting easily to change. Data Vault modeling does a great job of addressing this requirement. But even so data vault can be said to apply the 80/20 rule when it comes to adaptability.
When a new attribute is introduced to a data vault it may be added as part of a new satellite or it may require us to modify an existing satellite. With Anchor this will always be a new table per attribute. But to classify Data Vault as cumbersome in this category is not a fair assessment.

- **Immutability.**

This is in fact a very important distinction between Data Vault modeling and Anchor. It is true that Data Vault considers the natural business key to be immutable. Note however that this is in the object-oriented view of immutability (cannot be modified after it is created) versus the general definition (not subject to change).

While natural business keys should not change its composition, we have come to realize that all things can of course change. If a new natural business key is created for an existing instance of a concept then a new Hub instance is created in a data vault model (with a same-as-link or SAL to let us know the keys point to the same concept). With Anchor, the new natural business key can live alongside the old one as a natural key view of that domain (concept). The Anchor itself is keyed by a sequence ID that remains the same (immutable identities).

- **Natural to Surrogate.**

The natural key to surrogate is 1:1 in the DV and M:1 in Anchor. In Data Vault a natural key that has changed (over time) = a new key = a new surrogate. With Anchor a natural key that changes = a second (historical) reference to the same surrogate (still only one surrogate).

- **Assumptions.**

Both Data Vault and Anchor are indeed built to change. Each are forms of Ensemble modeling and take advantage of unified decomposition to split out the things that change from the things that don’t. This entry could be modified to read “built to change – some assumptions” for Data Vault.

**Comparison Summary.**

Hans and Lars determined that the Data Vault and Anchor modeling patterns are very similar in many ways. Most important factors here include:

- Focus on business defined schemas
- Core Business Concept / Domain
- Unified Decomposition
- Models for adaptability & history
- Concept instance in Hub or Anchor

The differences are somewhat structural and some others are in the deployment. The most important differences include:

- DV context satellites = logical attribute groupings vs. Anchor single Attributes
- DV uses load d/t vs. Anchor effective d/t
- DV Hubs (Surrogate & Natural key) vs, Anchors (Surrogate only)
- DV Natural key immutable vs. Anchor not
- DV composite keys = concat bus key vs. Anchor dynamic natural key views

Ultimately the modeling patterns are closely related in the same family of Ensemble modeling forms. Focusing on the common features with a thoughtful process to select the flavor can be considered best practice.
Lars Rönnbäck
lars.ronnback@anchormodeling.com
www.AnchorModeling.com

Hans Hultgren
Hans@GeneseeAcademy.com
www.GeneseeAcademy.com