Modeling Pattern Characteristics 2

Analyzing Modeling Pattern Characteristics & Approaches II

GENESEE ACADEMY, LLC

2013

Authored by: Hans Hultgren
Index

INDEX .....................................................................................................................................1
FORWARD................................................................................................................................2
THE MODELING APPROACHES & THE CHARACTERISTICS.........................................................2
DATA VAULT MODELING..............................................................................................................3
  Decomposed | Encapsulated ..........................................................................................................3
  Business Concept | Abstracted ........................................................................................................3
  Attributed | Generic ....................................................................................................................4
ANCHOR MODELING................................................................................................................5
  Decomposed | Encapsulated ..............................................................................................................5
  Business Concept | Abstracted ..........................................................................................................5
  Attributed | Generic .....................................................................................................................6
3NF NORMALIZED MODELING...................................................................................................7
  Decomposed | Encapsulated ..............................................................................................................7
  Business Concept | Abstracted ..........................................................................................................7
  Attributed | Generic .....................................................................................................................8
DIMENSIONAL MODELING.......................................................................................................9
  Decomposed | Encapsulated ..............................................................................................................9
  Business Concept | Abstracted ..........................................................................................................9
  Attributed | Generic .....................................................................................................................10
MORE INFORMATION...............................................................................................................11
APPENDIX: CHARACTERISTICS............................................................................................12
**Forward**

This document picks up where the first installment left off. The first document presented the overview of modeling pattern characteristics, the primary three characteristics for this analysis and then proceeded to map these characteristics against the first set of data modeling approaches. To recap, the three characteristics include: a) Encapsulated versus Decomposed table patterns, b) Abstracted versus Business Concept orientation, and c) Generic versus Attributed context. The definition of the three pattern characteristics is defined again in the Appendix of this document.

**The Modeling Approaches & the Characteristics**

In this next installment the first set of modeling approaches is further analyzed. These include Data Vault Modeling, Anchor Modeling, 3NF Normalized and Dimensional Modeling. In this paper the modeling approaches are analyzed in each category with an explanation of how and why they fit each category.
Data Vault Modeling

Data Vault Modeling is the leading Ensemble modeling method. A Core Business Concept (Customer, Employee, Product, Sale, etc.) is represented in the model as a unified grouping of component parts. These parts together serve to register, define, integrate and historize unique instances of the business concept.

**Decomposed | Encapsulated**

Looking to the modeling pattern characteristics, data vault modeling makes a move towards **Decomposed** since concepts are represented as groupings of component parts. The parts of a concept include Hubs, Links and Satellites. In practice this means that a core business concept would require at a minimum 2 parts (Hub and one Satellite). In most cases there will be more parts as there will be relationships (Links) and more than one Satellite (designed by rate of change, subject area, type of data or source). Though data vault modeling does practice the Unified Decomposition common to Ensemble modeling methods, it is not fully decomposed and so it offers a great deal of flexibility concerning the extent of decomposition. These decisions are made by the modeler/architect who is developing the particular data model. The number of Satellites for a given Hub can range from 1 to 20 (or more). In practice there might be somewhere between 2 and 7 on average.

**Business Concept | Abstracted**

Data Vault Ensembles are commonly based on Core Business Concepts. The idea of Abstracted concepts is not typical in data vault based EDW programs. So the concepts are likely to be Customer, Vendor, Employee, Product, Sale, Order, Return and not likely to be Party, Thing, or Event. There are those who use vault modeling patterns with more abstracted forms but this is not the standard for data vault deployments.
Attributed | Generic

The Data Vault modeling approach is based on the Attributed paradigm.

Attributed models are schema-based models that specifically identify the context attributes required to define a business concept or relationship. So when describing the context for a Customer for example, the schema would specify the attributes:

- Salutation, F_Name, L_Name, M_Init, Credential, Add_1, Add_2, City, State,
- Postal_Code, State/Province, Country, W_eMail, P_eMail, W_Phone, H_Phone,
- M_Phone, B_Day, B_Cntry, Language, Ed_Lvl, Cred_Score

Generic forms would step away from defining these specific attributes and instead deploy a form of Name-Value Pair (NVP). So the attributes in model would be:

- Name, Value

In this generic form, a record (the context for an instance of a business concept) is represented in several rows. There is one row required for each context attribute. Compared to the Attributed example above, there would be 22 rows versus 1 row. This is why the generic forms are sometimes referred to as vertical (and the attributed forms as horizontal).
Anchor Modeling

Anchor Modeling is another form of Ensemble modeling that utilizes a more highly decomposed paradigm for context attributes. A Core Business Concept (Customer, Employee, Product, Sale, etc.) is represented as an ensemble in a similar way to data vault modeling.

Decomposed | Encapsulated
Anchor modeling represents perhaps the most decomposed form possible. This extreme (or total) decomposition is achieved by breaking out all context attributes into their own table forms. One Attribute = One Table. Arguably this is the purest logical form of normalization for an EDW where all history is tracked. Since any attribute can theoretically change (or be added) independently of any other attribute, each attribute should be tracked and managed separately.

Business Concept | Abstracted
Anchor modeling is also based on the Core Business Concept. At the heart of this concept is an Anchor that serves a similar center of the concept purpose as a Hub serves in Data Vault modeling.

Abstracting concepts in an Anchor model would largely defeat the purpose and the utility of this form. Since the attributes are effectively fully flexible at the same time as they are specifically descriptive, moving to an abstracted form would be trying to solve a problem that has already been solved.
**Attributed | Generic**

Anchor modeling is also an Attributed modeling form. Interestingly it is both the furthest away and the closest to generic context forms. It is the furthest away because it creates a schema-based, horizontal model that has architected table forms for every attribute. It is the closest form in that it is one short step away from NVP. Each Attribute table has logically only one Attribute (the specific context attribute name) and the records contain row values for that one attribute only. Anchor is the closest point to generic attribution while remaining a fully attributed context model. To use generic attribution in the Anchor model would just be a Generic Model (no longer a form of Anchor modeling).

Foundationally Anchor modeling can be seen as the extreme point in the attributed modeling world when it comes to addressing the flexibility and non-destructive requirements of the agile data warehouse. The next level of agility would require a move to a generic form.
3NF Normalized Modeling

The world’s most common form of data modeling is third normal form (3NF) modeling. This modeling approach is based on modeling the Persons, Places, Things and Events that define our organization and then relating them to each other based on the specific cardinality of business-rule driven relationships. Referential integrity is established through key constraints among and between the tables in the database in order to support the accurate capture and management of operational data.

Decomposed | Encapsulated

3NF Entities (concepts) are encapsulated based on a single grain of key. This means that the context that describes a business concept (including the defining relationships) is embedded in a single common table (Entity). So for example a Customer Entity includes all things that describe the core concept of Customer. This Entity would then be represented a table in the database.

Business Concept | Abstracted

Because the Entity in a 3NF model is typically in line with the idea of a concept, this modeling form is generally based on Business Concept. However in 3NF there is often a practice of applying some levels of abstraction in the form of Super-Typing. So for example there might be an abstraction to Person that would be used in place of Employee, Customer and Consultant. Some of this is driven by the perceived need to reduce table count and record redundancy. Other times this may be for reasons of establishing master records for common concepts. And lastly this might be an abstraction driven by an alignment with central business meaning (such as with information models, industry reference models, and forms of taxonomies, etc.).
Note however that these abstractions are not innate to 3NF modeling. When these abstractions are used in a data warehousing setting we classify the modeling form as a 3NF Generic modeling form.

**Attributed | Generic**

At the foundation of a 3NF Entity is a set of descriptive context attributes. The 3NF entity is a specific schema-based model (horizontal) wherein each Entity contains all Attributes that depend on the key of that Entity. The diagram would indicate that this form is fully attributed except that there is a tendency in 3NF to make heavy use of codes (mainly for reference tables). For example, instead of including a Customer Size attribute (“Large”), we use a Customer_Size_Code (“102”). There is then a relationship via foreign key (FK) in the Entity pointing to a primary key (PK) of a Customer_Size table. When we join to this table we can lookup the code 102 and find that the corresponding descriptive value is Large. This phenomenon is a byproduct of way we apply 3NF models – to manage the business-rule-driven accurate capture of data. In these cases we are looking to limit the possible values of Customer Size to a pre-determined and managed set of value (Domain Values). This helps use to avoid data anomalies and also helps us with categorizing and classifying our data.

Note that the reasons for using these codes and related tables are not related to data warehouse governance (but rather operational governance). In the data warehouse we are taking copies of data that has already been captured and so these controls are not required.
**Dimensional Modeling**

The idea of dimensional modeling is to present a set of tables (Dimensions) that each include all relevant descriptive information about a concept (Customer, Store, Product). These dimensions are used in conjunction with Facts (metrics, measures, amounts, aggregations, counts, etc.) that allow us to view and explore (drill up, drill down, pivot, graph, etc.) these Facts by the related Dimensions. So a data mart might include Sales by Customer, by Product, by Store, by Date, etc.

**Decomposed | Encapsulated**

The Dimension should ideally include all descriptive information in a single table form. For this reason the Dimension sometimes includes data from outliers (related tables that further describe the concept but at a higher level grain). In data warehousing the idea of Conformed Dimensions means that the dimension should mean the same thing to all Facts where it is applied. For this reason, the Dimensional Modeling approach applies almost fully Encapsulated table forms.

Although the form itself does not promote decomposition, it is possible to manage some decomposition by breaking out large dimensions into mini-dimensions. Though not Unified Decomposition (ensemble with a common shared key), these mini-dimensions can be associated with each other using a form of factless fact.

**Business Concept | Abstracted**

The Dimension is approximately based on a core concept. With outliers rolled in and the possibility of partial sets of context, there are exceptions to the pure alignment with a Business Concept. However it is possible to deploy Dimensional modeling and base all the Dimensions on the Core Business Concept.
In either case the paradigm of Abstracted concepts is not a component of Dimensional modeling. If the dataset feeding the Dimensional presentation layer model (data mart) was in an abstracted form, the process of loading the data mart would include the logic to move back to the core concept (Thing of Type Product, and Party with Role Play of Customer would become Dim_Product and Dim_Customer in the star schema).

**Attributed | Generic**

Since the primary usage of dimensional models is in the presentation layer (Data Marts) the data contained in this layer must be relevant and understandable by the business. The Dimensions need to contain the descriptive business context related to each concept. For this reason there is no place for generic attribution in dimensional modeling. By design Dimensions need to be descriptive, modeled schemas with attributed (horizontal) context.
More Information

For Seminars, Training and Certification on Data Modeling, Data Vault, and Data Warehousing please see GeneseeAcademy.com

For online training including updates and current topics please visit us online at DataVaultAcademy.com
APPENDIX

Characteristics

I. Encapsulated versus Decomposed Table Patterns

Encapsulated table patterns have been the standard in data modeling for the past 30 years. Today the primary forms are third normal form 3NF and dimensional modeling. In each of these forms a core business concept is represented by an Entity (or dimension) that includes the business key and all context attributes within the same table.

Decomposed table patterns. As is now common for data warehouse modeling, there are patterns that practice forms of table decomposition. This means that the business keys and context attributes that define a core business concept are translated to a set of interrelated tables. These tables are parts of the whole – an Ensemble – and together they represent the same information as would be in the encapsulated table forms.

In this diagram we see an Encapsulated table form (Entity) next to a Decomposed table form (Ensemble). An Ensemble complies with Unified Decomposition which means that all parts are considered together and only have meaning in relation to the whole.

II. Abstracted versus Business Concept Orientation

Abstracted concepts are forms of super-typed or high-level entities. In effect these are classifications of things found at the top levels of taxonomies or hierarchies. So a customer, vendor and employee are all types of party at an abstracted level. We tend to use abstracted concepts primarily for information modeling and in that way they are often found in industry reference models as well. There are those who also use abstracted concepts in data modeling (logical and physical).
Business Concepts, or Core Business Concepts, are the business-driven, natural levels of entities. They represent the level at which the business actually transacts, creates, discusses, uses and reports on these entities. So customer, vendor and employee are common business concepts.

This diagram illustrates Business Concept level in comparison to the Abstracted level. Notice that the level of concept is determined by the frequency of actual usage in the business (the number of times the concept is referenced in business).

III. **Generic versus Attributed Context**

**Generic** Context refers to generic forms being applied in capturing and storing the descriptive information about the concepts in our data warehouse. Commonly the generic form is name/value pair (NVP) although it can also be other forms of n-structured data. So the data stored without a defined and modeled schema (attributed data model). Records/Rows in these tables include two parts 1) the name of the attribute / or tag, and 2) the corresponding data value for that instance. Example would be a table that has effectively two attributes: name & value. The records within this table then define the attribute name matched with each data value (fname:Hans; lname:Hultgren; email:hans@edwi.org and etc.). This approach is sometimes referred to as data-driven or vertical.

**Attributed** Context refers to actual modeled schemas that include specific attribute names in the data model. This is the traditional approach to data modeling where the context that defines the entity or concept is analyzed,
designed and modeled into the table structure. This approach is sometimes referred to as model-driven or horizontal.

Notice here that the NVP pattern above has effectively only two attributes, Name and Value. This form does not communicate the context attributes that are planned (anticipated, expected, or required). To understand what types of context attributes might exist we need to query the data in the table.

In the Attributed model above right we can see that the schema itself communicates the specific attributes that have been designed into the model. In this case the model itself communicates the specific context attributes that exist.