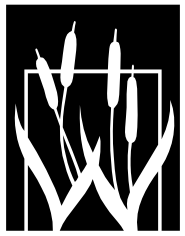


Whitemarsh
Information Systems Corporation

A Column by Any Other Name
Is Not
A Data Element

Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716
Tele: 301-249-1142
Email: Whitemarsh@wiscorp.com
Web: www.wiscorp.com

Table of Contents

Objective of Talk

- Describe an approach to achieve enterprise-wide data standardization
- Through the specification, implementation, and maintenance of data elements
- Within the context of a metadata-repository, CASE-like environment

Fundamental Data Element Definition

- Data elements are context independent business fact semantic template
- That are employed to fully define and control context dependent business facts
- Such as attributes of entities, columns of tables, fields on forms, etc.

Test of this Approach

- "Does this approach make common sense?"
- Is the demonstration compelling?
- If yes, then consult references for the much deeper presentations

A Michael Brackett Story from DAMA 2001

- Michael Brackett sat in on the my Enterprise Wide Data Standardization presentation at DAMA 2001 in Anaheim.

- I "creatively acquired" significant components of the Whitemarsh approach from a May 1995 Michael Brackett talk

- It was sort of like having Michelangelo sit in our your one-person art show.

- You're both wanting his review and critique but are deathly afraid that he will give it, or even worse, just walk out half way through. I got the critique and review, and Michael stayed to the very end.

- After the presentation was over, a meeting with Michael produced the observation that the approach was fundamentally sound but that he disagreed with my assertion that an enterprise only had 2000 data elements. Wow, I thought, he agrees with the approach. That's great!

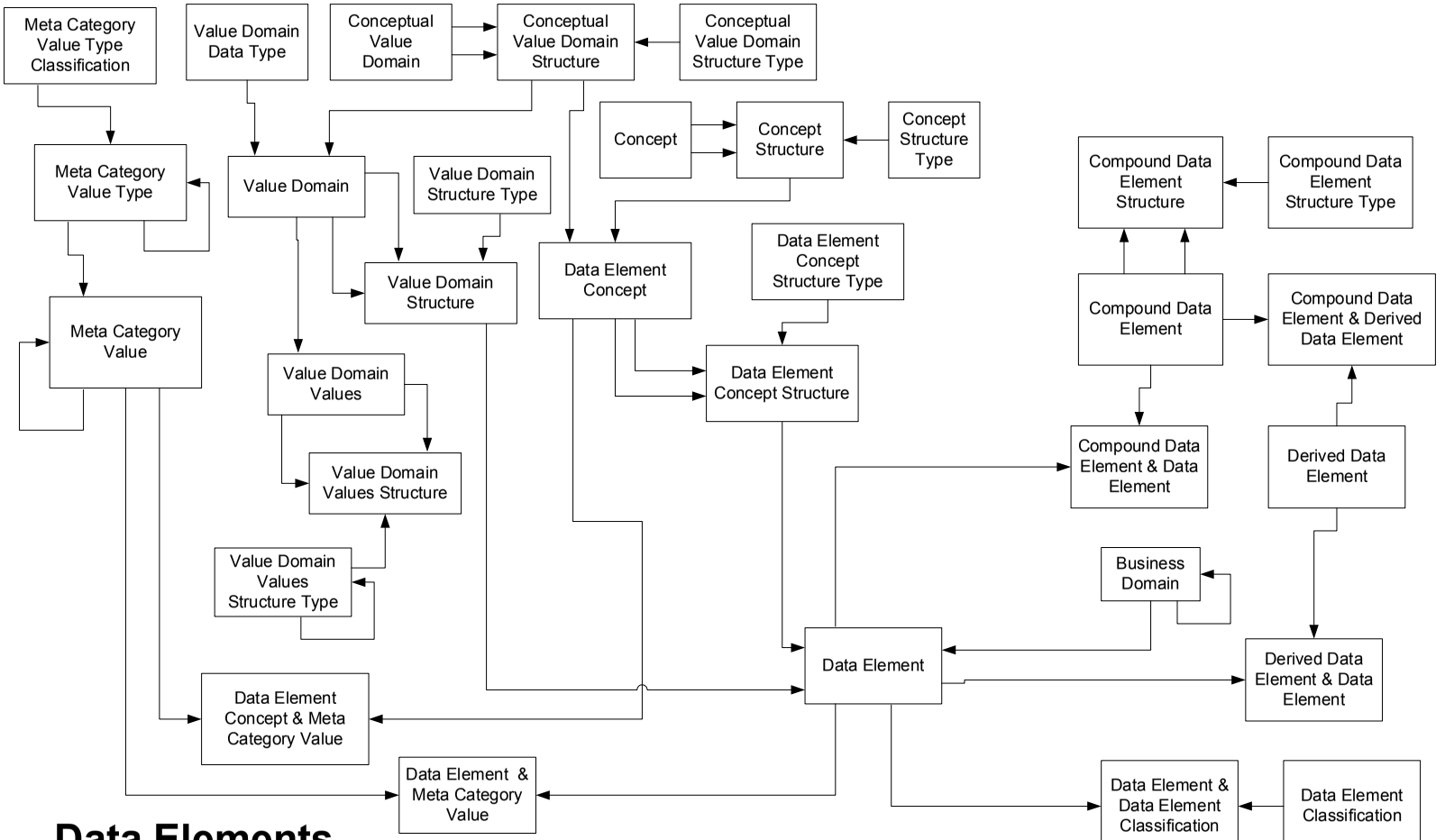
- Brackett stated that the state of Washington, for example, has about 20,000 data elements. A casual review of a "business" like the State of Washington is between 10-15 different enterprises (Agriculture, Education, Environment, Justice, Transportation, Welfare, etc.)

The Approach

- Create a re-usable cache of data elements
- Establish a metadata repository and CASE environment
  - Define database columns,
  - Interrelate the defined columns through the data element metadata, and
  - Support both forward engineering of new databases and reverse engineering of existing databases.

Brief Review

- A data element is a context independent business fact semantic template.
- An amalgamation of familiar concepts expressed as single words brought together under a single name
- The name is not the data element, the collection of semantics is. The name is merely a discrete value-based alternative representation of the semantics.
- Critical to reuse is a metadata model that exists within a repository type database.
- Finally, since all data elements are not just elementary atomic fact templates, both compound facts, and derived facts must be represented.



Five collections of meta entities[1]

- Semantic Hierarchies (meta category value entities)
- Concepts, Conceptual Value Domains, Data Element Concepts and Value Domains
- "Fully Crested" Data Element
- Compound and Derived Data Element
- Data Element Classification Schemes

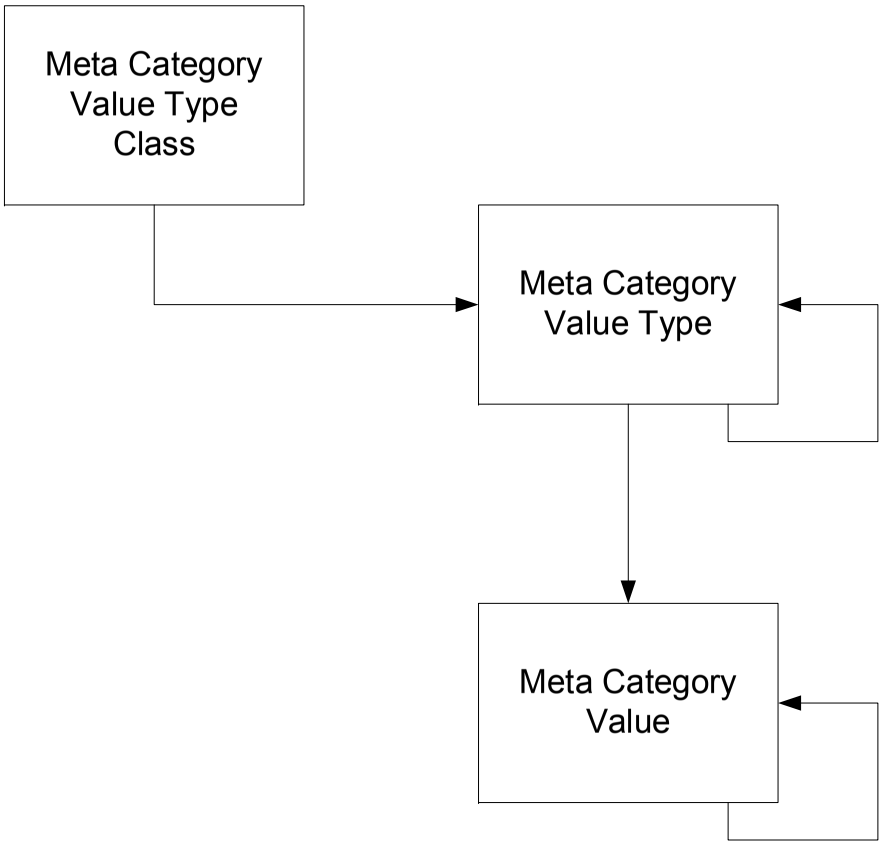
Key Michael Brackets observations:

- Data elements exist within natural classification hierarchies

- Data element names are comprised of well known business domain words

Semantic Hierarchies: Meta Category Value and Meta Category Value Types

1 *Other meta entities are needed to make this "relational" data element model compliant with 11179 Part 3*



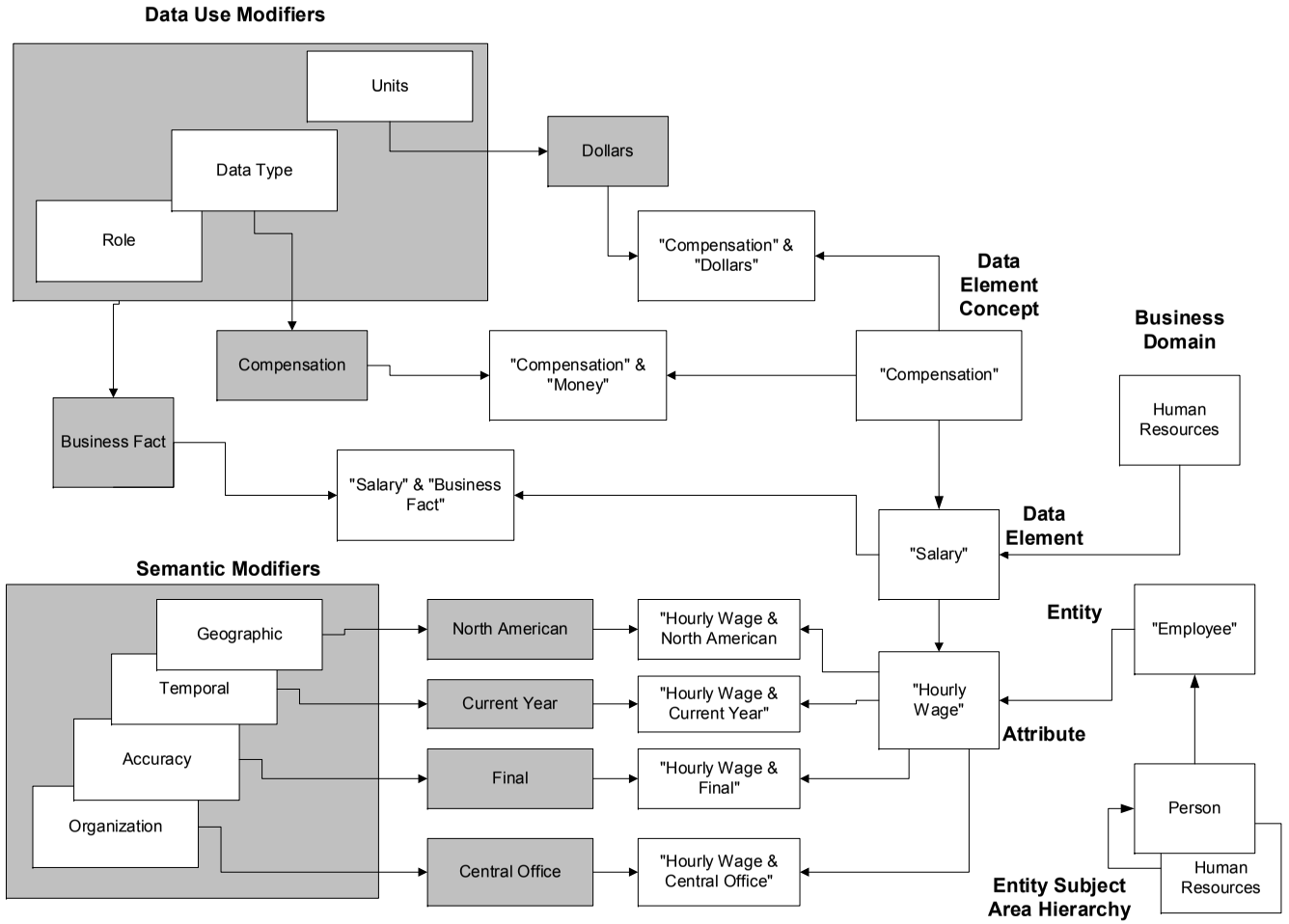
Examples of Prefix Components for a Data Element

Meta Category Value Type and Meta Category Value Hierarchies				
Meta Category Value Type Hierarchy	**Examples of Semantic Modifiers**			
	Temporal	**Accuracy**	**Geographic**	**Organizational**
Examples of Meta Category Value Hierarchies for the Meta Category Value Types	Last	Estimated	World	World-wide
	First	Projected	Hemisphere	Business unit
	Latest	Revised	North America	Region
	Earliest	Initial	United States	District
	Current	Actual	Mid-Atlantic	Territory
	This year		Maryland	
	Last year		Bowie	

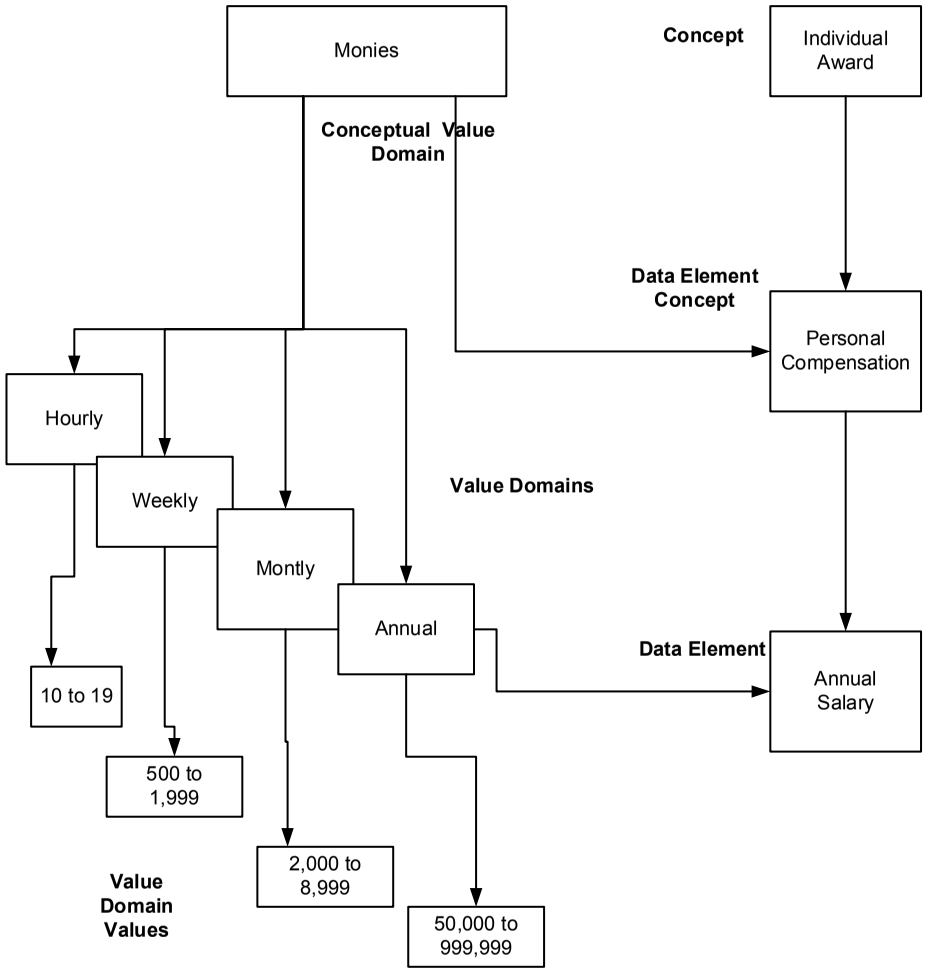
Examples of Suffix Components for a Data Element

Meta Category Value Type and Meta Category Value Hierarchies			
Meta Category Value Type Hierarchy	**Data User Modifiers**		
	Data Type	**Role**	**Unit**
Examples of Meta Category Value Hierarchies for the Meta Category Value Types	Date or date component	Identifier component	Day
	Code	Factor	Case
	Text	Flag	Aisle
	Weight	Indicator	Pallet
	Dimension	Identifier component	Transaction
	Money	Rank	Percent
	Integer	Business fact	Inches

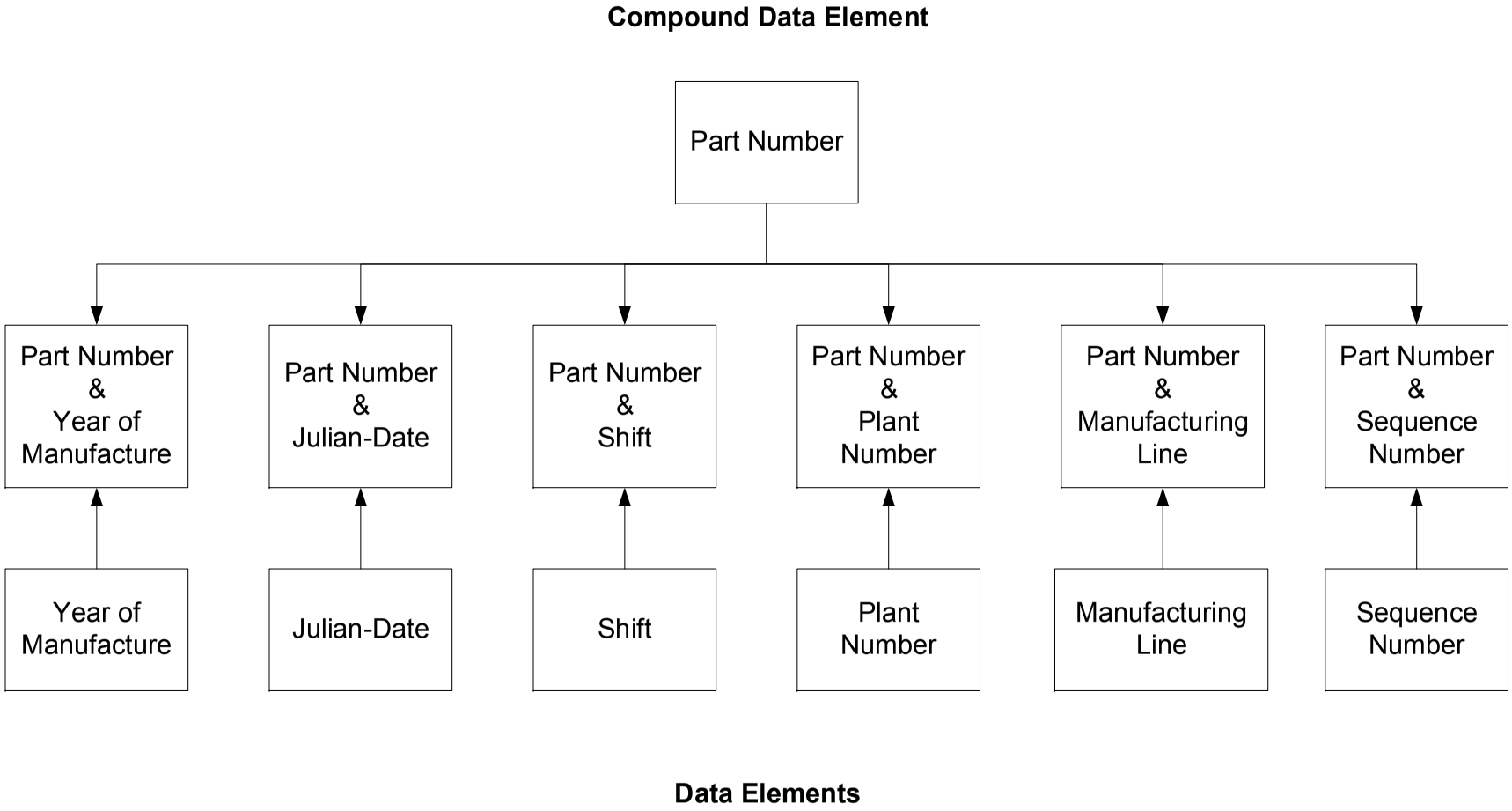
Example of Complete Metadata for a Data Element



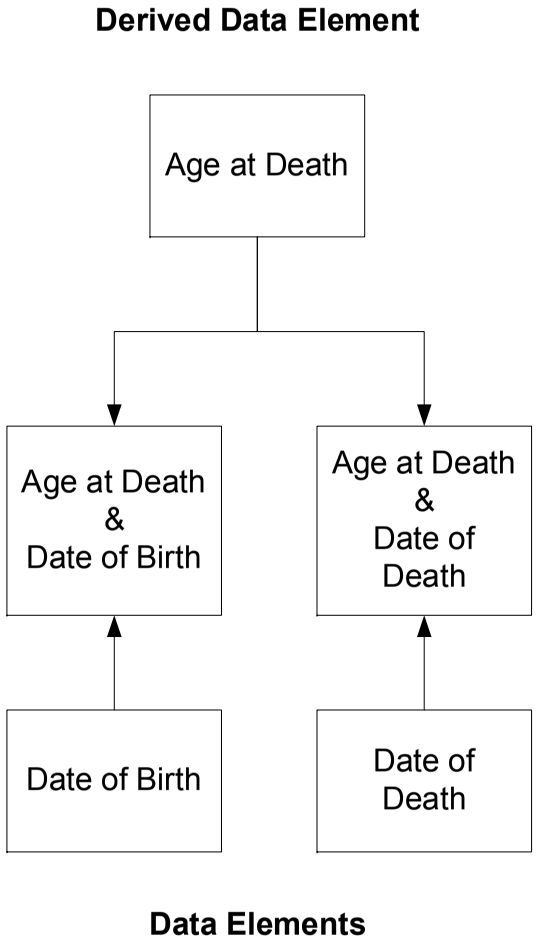
Concepts, Conceptual Value Domains and Data Element Concepts



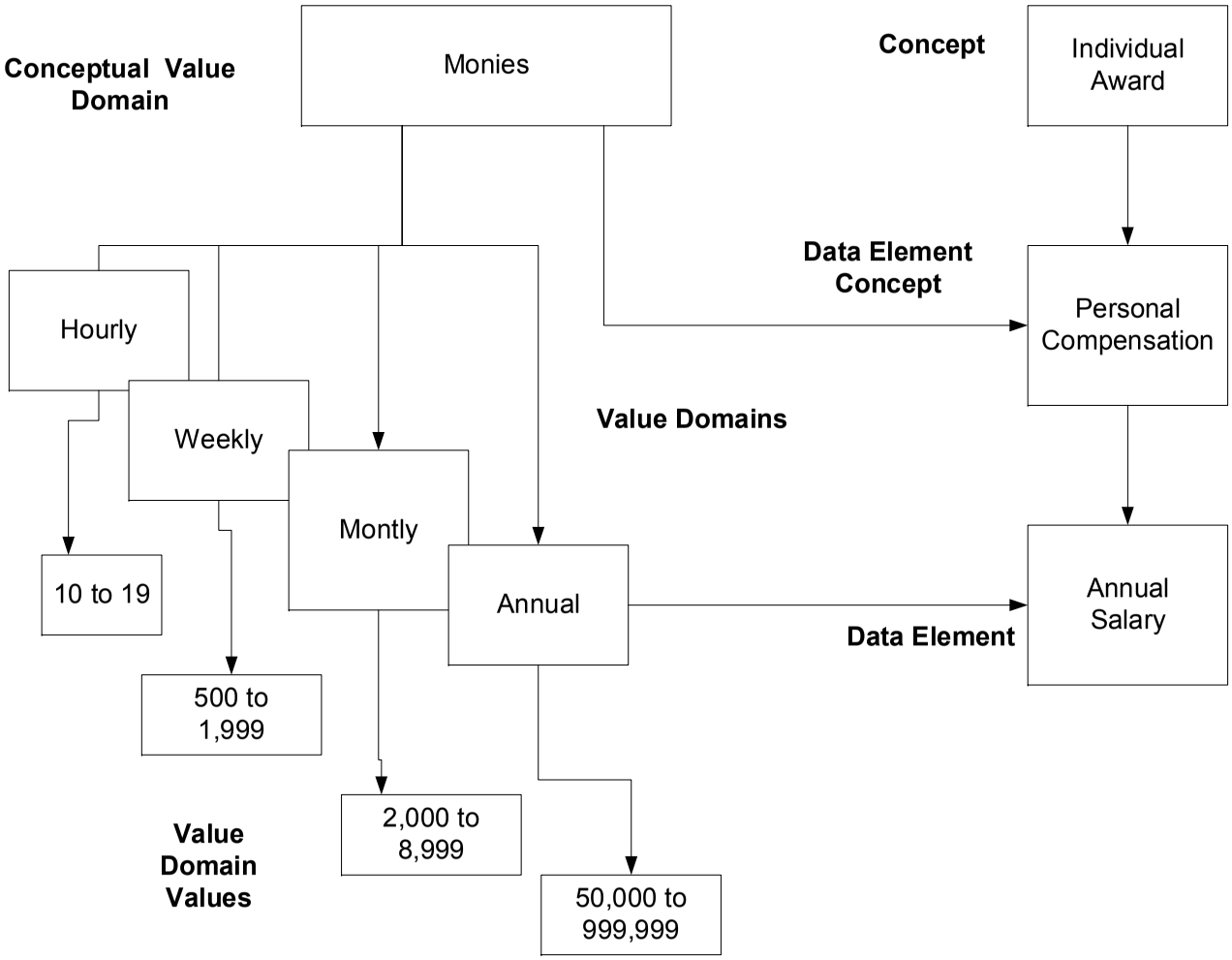
Compound Data Element

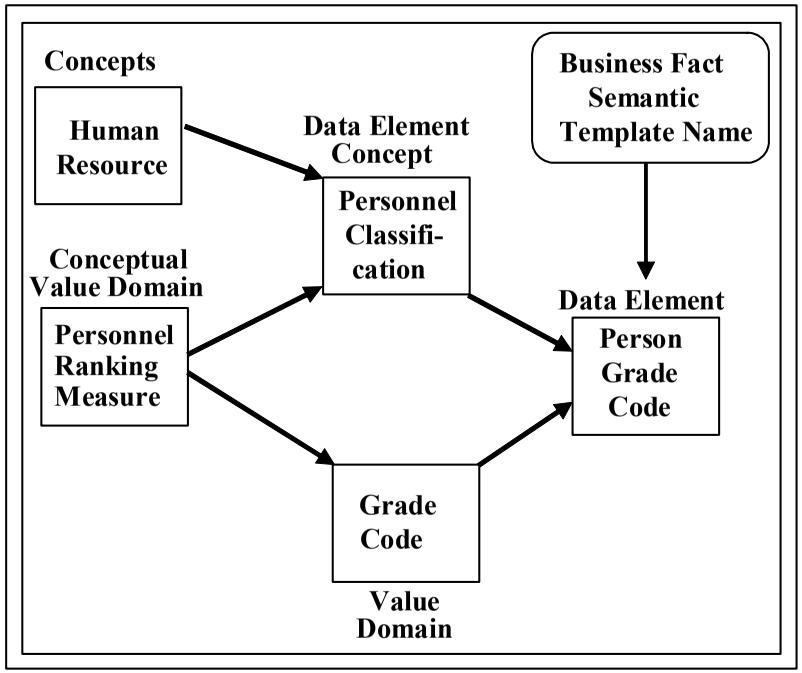


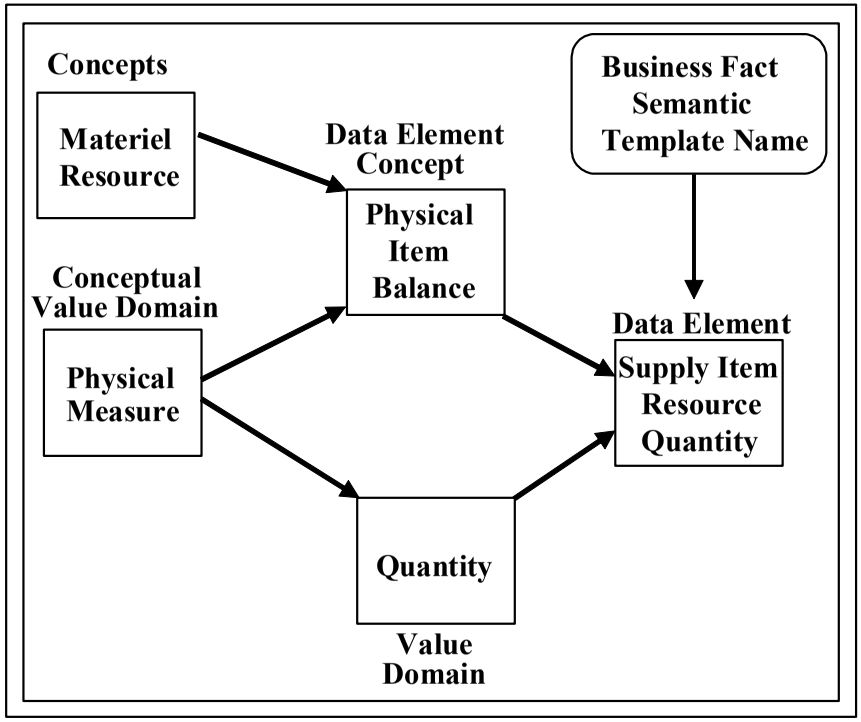
Derived Data Element



Value Domains







Benefits to this Approach

Common statistics about database environments for a typical state government, or multiple business line enterprise.

Unit	Components
1 database	100 tables
1 table	15 columns
Each agency	100 databases
Each state	40 agencies

- The total count of columns for each database is 1500.
- The total columns for an agency is 15,000 columns,
- The total for the state, for example, like Washington, New York, or Maryland would be 600,000.

Thus, if Michael Brackett's assertion is true, then each of the asserted 20,000 data elements is reused about 30 times.

Alternative Approaches and Cost Comparisons		
Data Standardization Alternative for a multi-site, multi-application Government MIS	**Final Quantity of columns, fields, cells, etc.**	**Cost via technique employed for definition**
Accomplished traditionally (prime + modifier + classword) across all systems	19,000	$6.75 million
Alternatively, if accomplished by standardizing closely named columns and fields	3,000	$1.06 million
Alternatively, if accomplished through Comprehensive Data Standardization Techniques–Eliminates redundant–but different-- representations of the same concept	560	$200,000

In this example the ratio was 34 to 1.
Another Example: US Department of Defense Agency ETL Effort

- Each: requirement, design, software implementation and maintenance.
- Each ETL represents a failure in data standardization.
- Columns supposed to be the same have different names, semantics, data types, levels of granularity, time-sequencing, and the like.
- While an enterprise-wide data element standardization approach would not solve all these problems, it would clearly affect different names, semantics, data types.
- The agency spends about $175,000,000 each and every year on such ETL activities.
- If the data element approach resolved 50% then that would represent savings of about $90 million per year.
- Extended to the US DoD as a whole the savings would be about $450 million, and to the U.S. Government as a whole, about $1.5 Billion.
- Given that the US Government spending represents about 10% of the total economy, then the savings to the economy as a whole is about $15 Billion.

Final Example: Data Element Value Domain Synchronization

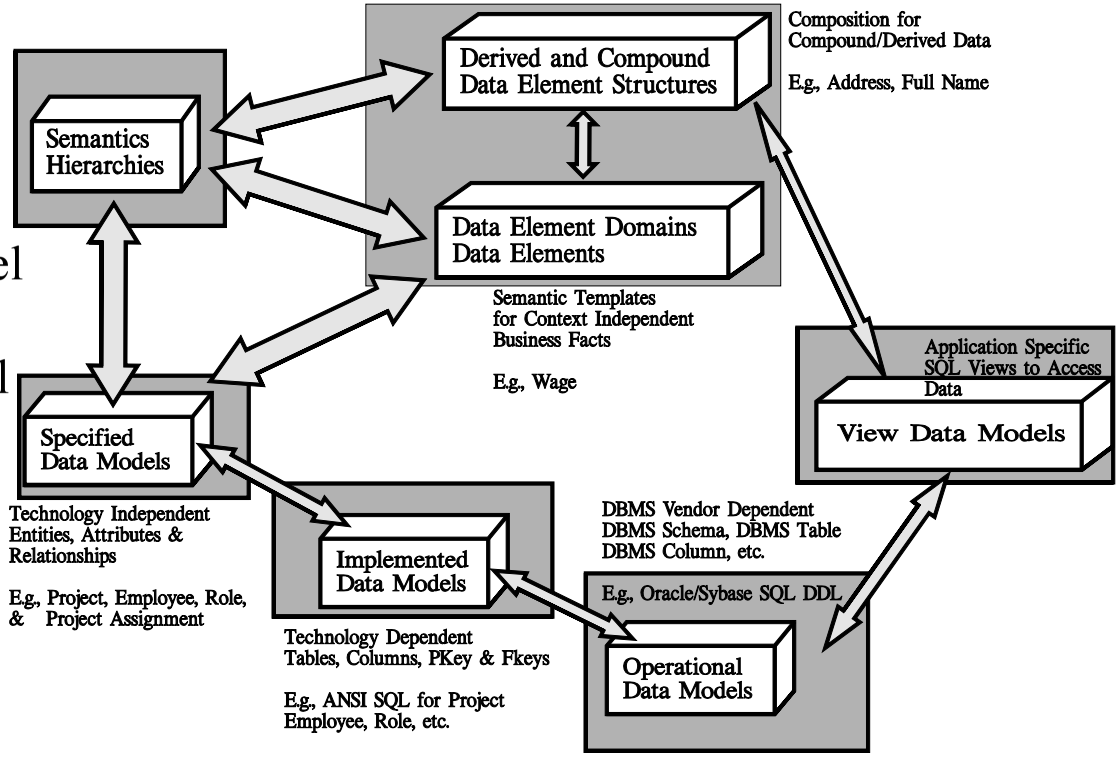
Large number of different SQL tables within a health care environment that had columns that required a "yes or no" answer. Here's the value distribution.

Yes|No Column Frequency	Code values		Yes|No Column Frequency	Code values
1296	1 = yes…..0 = no…..		8	1 = yes…..0 = no…..99 = unknown
899	Y = yes…..N = no…..		1	1 = yes…..0 = no/negative…..99 = unknown
740	1 = yes…..…..		1	1 = yes…..0 = no…..2 = unknown
75	Y = yes…..…..		24	Y = yes…..N = no…..NA = not applicable
17	0 = yes…..1 = no…..		17	Y/N items to be deciphered…..
2	1 = yes…..0 = no…..2= error		11	1 = yes…..0 = no…..2 = not applicable
104	Y = yes…..N = no U = unknown		1	1 = yes…..0 = no…..U = undetermined
1	Y = yes…..N = no…..D = does not apply		1	1 = yes…..2 = no…..3 = unknown
3	0 = false…..1 = true…..		2	Y = yes…..N = no…..blank = no
5	Y = yes…..N = no…..N/A = not applicable		1	0 = yes…..1 = no…..2 = not applicable
90	1 = yes…..2 = no…..		1	2 = yes…..1 = no…..
6	1 = yes…..2 = no…..3 = not applicable		1	1 = yes…..0 = no…..U = unknown
1	1 = yes…..2 = no…..N = none		1	Y = yes N = no A = ask
2	1 = yes…..2 = no…..0 = do not use default		1	Y = yes N = no R = restricted

Achieving the Benefits of Data Element Standardization within the Database Environment

Integrated CASE-Repository environment.

- Semantic Hierarchies
- Data Elements
- Specified Data Model
- Implemented Data Model
- Operational Data Model
- Application View Model



Specified Data Models

- Data model templates. These represent commonly used collections of entities, attributes, and relationships that are organized by subject areas.
- Each entity is a coherent policy-based collection of attributes
- Semantic hierarchies from the data element meta model are mapped to attributes
- Specified data models are not database designs. Its paradigm is subject, entity, attribute. Relationships connect various entities, and can related entities across subject areas.

Benefits from Specified Data Models

Because of these work saving assists, creating a specified data model is an effort that is characterized by:

- Significantly shorter times to create entities because of all the inherited semantics
- Lowered risk because when things are the same they will be semantically defined the same
- Increased quality because there will be more time for important semantic component parts
- Increased productivity because the process of creating the specified data model can be accomplished by functional experts rather than just data administration staff.

Implemented Data Models

- Database data models exist in two necessary forms: DBMS independent and DBMS dependent.
- Both these models are needed because databases that are actually deployed may have table designs changed to meet the needs of different DBMSs, operating systems, data architecture classes, or computer hardware capacities.
- In such cases, the "real" database design is not intrinsically changed, just deployed differently. Consequently, the DBMS independent database designs are needed.
- DBMS Independent database data models are thus database data model templates for operational database data models deployed across the enterprise.
- The triple of the implemented data model are schema, table, and column.
- Data elements are mapped to columns. Attributes are mapped to the columns.

Benefits from Implemented Data Models

- The implemented data model is distinct from the specified data model.
- It may be the technology dependent transformation of a collection of entities from within the specified data model.
- The implemented data model may have a multiple subject-area scope and thus may represent interrelated collections of entities, attributes and relationships from the different subject areas.
- Conveys COTS (commercial off the shelf) package vendor's data model in a form that is understood by the rest of the enterprise.
- The implemented database represents databases to be implemented on some technology dependent platform. That is, through one or more DBMSs and one or more specific computers

Operational Data Models

- DBMS data models for the operational set of schemas that exist within specific hardware, operating systems, and DBMSs.
- The triple is DBMS schema, DBMS table, and DBMS column. DBMSs columns are mapped to columns from the implemented data model tables.
- Because of the divergence of DBMS vendor implementations of SQL are different operational data model variations of the same implemented data model.
- The operational data model is not only DBMS specific, it is also targeted to a specific operating environment.
- Thus, there can be multiple operational data models, each with a somewhat different design due to DBMS characteristics and performance requirements for every database.
- The data models contained in the data structures within the specified data models could appear in multiple implemented and operational data models.

View Data Models

- Application view data models consist of views and their view columns with the associated hierarchies.

- View elements map to DBMS columns and as appropriate compound data elements and derived data elements.

Approach Summary

- Data elements mapped through attributes of the specified data model and/or through columns of the implemented data model enable fact based semantic homonyms regardless of name changes.

- Semantic hierarchies attached to data elements are mapped to attributes or columns enable the identification of semantic homogeneous or related context dependent business facts that exist ultimately as DBMS columns.

- Specified data model templates serve as templates for complete or partial tables within the design of databases.

- Implemented data model collections of schemas, tables, and columns can be employed to deploy different operational data models

- DBMS data model columns are mapped to application view columns.

- The complete integration of these models, that is, semantic hierarchies, data elements, specified data models, implemented data models, operational data models and application view models finally gives enterprises the metadata through which integrated, shareable, enterprise-wide data standardization can be achieved.