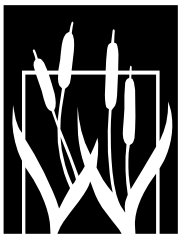

Whitemarsh
Information Systems Corporation

Whitemarsh Metabase Data Modeler: Data Element Users Guide

December 2007

Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716
Tele: 1-301-249-1142
Email: Whitemarsh@wiscorp.com
Web: www.wiscorp.com

Table of Contents

List of Figures

1 Introduction

The data element component of the data modeler module is designed to capture data element specifications. That is, context independent business fact semantic templates. Fully supporting data elements are semantic hierarchies, conceptual value domains value domains, data element classifications, data element concepts, compound data elements, and derived data elements.

Collectively these data element semantics support the complete understanding of these context independent business fact templates so that they can be efficiently and effectively employed for attributed within entities and as columns within tables.

The document, *Data Modeler Architecture and Concept of Operations*, which can be downloaded from the Whitemarsh website, www.wiscorp.com is an essential prerequisite reading for the correct use of this data modeler component. It presents the "business problem" being addressed. This user guide only briefly presents how to accomplish the solution.

Presumed Knowledge

This user guide, and all the other metabase user guides presume that the reader has read and is completely familiar with the following documents: Metabase Common Processes, and Metabase Bill of Materials and Single File Recursion (BOM/SFR Guide). These two documents serve as metabase teaching guides for processes that commonly occur throughout the metabase system.

The metabase supports spell checking through the Function key, F7.

Metabase Example

The metabase example, Movies, is a complete example of a business which is available from the Whitemarsh website. The Movies Rental Corporation was modeled after the largest movies rental corporation in the United States. As such, the example has national, regional, and retail outlets. There are two data models, one for an original data capture, store based system, and another which is a data warehouse for rented movies.

2 Software Installation

Metabase installation is explained in the Metabase Administrators Guide.

3 Database Design

The data element module has the following tables:

- Business Domain
- Compound Data Element

- Compound Data Element & Data Element
- Compound Data Element Structure
- Compound Data Element Structure Type
- Concept
- Concept Structure
- Concept Structure Type
- Conceptual Value Domain
- Conceptual Value Domain Structure
- Conceptual Value Domain Structure Type
- Data Element
- Data Element & Data Element Classification
- Data Element & Data Element Concept
- Data Element & Meta Category Value
- Data Element Classification
- Data Element Classification Structure
- Data Element Classification Structure Type
- Data Element Concept
- Data Element Concept Structure
- Data Element Concept Structure Type
- Derived Data Element
- Derived Data Element & Data Element
- Meta Category Value
- Meta Category Value Type
- Meta Category Value Type Classification
- Value Domain
- Value Domain Data Type
- Value Domain Structure
- Value Domain Structure Type
- Value Domain Value
- Value Domain Value Structure
- Value Domain Value Structure Type

The database design is depicted in Figure 1.

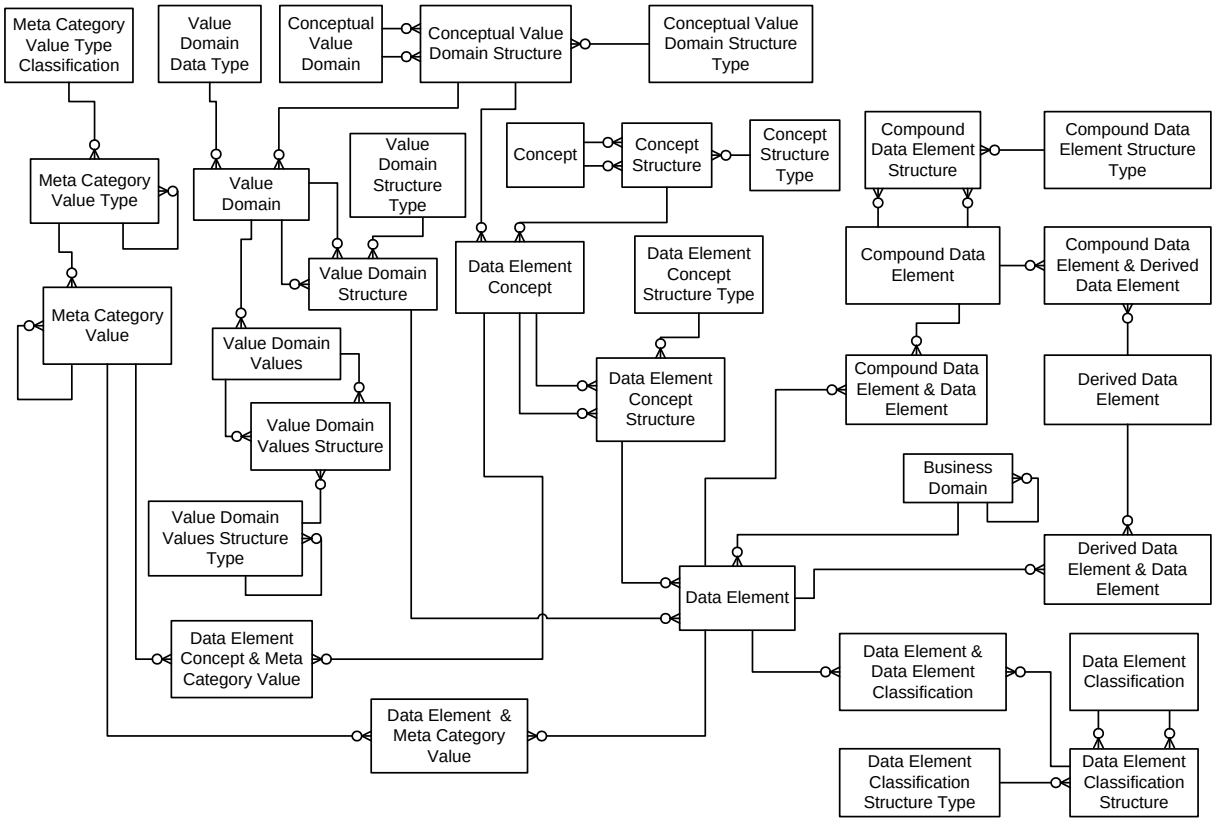

Figure 1. Data Element Metadata Model.

Explicit in this database design are the following:

- Business Domains are the parent of one or more data elements and enable data elements that are named the same to be distinguished by the business domain in which they reside.

- Compound Data Elements consist of multiple data elements in a specific sequence. Compound data elements may also consist of other compound data elements which too appear in a specific sequence

- Compound Data Element & Data Element represents the mapping between the one or more data elements and one or more compound data elements

- Compound Data Element Structures represents a collection of compound data elements that are in turn a compound data element.

- Compound Data Element Structure Type represents the classification of structure among a set of compound data elements.

- Concepts represent the sets of ideas, abstractions, or things in the real world that are identified with explicit boundaries and meaning and whose properties and behavior follow the same rules. Concepts are used as a basis for specifying the concepts of data elements.

- Concept Structures represent a collection of concepts that are related and are in turn represented by a collective concept.

Concept Structure Types represents the classification of structure among a set of concepts.

- Conceptual Value Domains represent the collection of concepts supporting value domains sets from which all value domains. A conceptual value domain forms a basis for a data element concept.

- Conceptual Value Domain Structures represent a collection of conceptual value domains that are related and are in turn represented by a collective conceptual value domain

- Conceptual value domain Structure Type represents the classification of structure among a set of conceptual value domains.

- Data Elements are context independent business fact templates. The complete set of semantics for a data element are those explicitly assigned, that is, meta category values, data element concepts, data element classifications, and business domains. It's inherited semantics include those assigned to its containing data element concept and value domains. The value domains of the data element are specifically assigned.

- Data Element & Data Element Classification represents the assignment of one or more data elements to one or more data element classifications.

- Data Element & Data Element Concept represents the assignment of one or more data elements to one or more data element concepts

- Data Element & Meta Category Value represents the assignment of one or more data elements to one or more meta category values.

- Data Element Classifications are schemes or ontologies to understand the knowledge categories within which data elements reside.

- Data Element Classification Structure represent a collection of data element classifications that are related and are in turn represented by a collective data element classification.

- Data Element Classification Structure Types represent the classification of structure among a set of data element classifications.

- Data Element Concepts represent the underlying concepts that support understanding the data element. A data element concept is the association of a concept and conceptual value domain.

- Data Element Concept Structure represent a collection of data element concepts that are related and are in turn represented by a collective data element concept.

- Data Element Concept Structure Type represent the classification of structure among a set of data element concepts.

- Derived Data Elements represent the result of some sort of calculation or transformation of either a collection of data elements or compound data elements or a combination of both.

- Derived Data Element & Data Element represents the assignment of a data element to one or more derived data elements.

- Meta Category Values are the individual semantics that are assigned to data elements in support of the complete specification of the data element. Meta category values are in turn collections within meta category value types. Meta category value collections can be hierarchically organized. The specifically assigned meta category value becomes an explicit part of a data element's name.

- Meta Category Value Type are collections of categories within which meta category values reside

- Meta Category Value Type Classification are broad classifications of the meta category value types and in turn, meta category values. There are two: prefix and suffix. The prefix meta category value type classification causes all assigned meta category values to appear before the common business name of the data element. Essentially, prefix semantics serve as a collection of "modifiers" to the base semantics of the data element. The suffix meta category value type classification causes all assigned meta category values to appear after the common business name of the data element. Essentially, suffix semantics serve as a collection "class words" that characterize the intended data type, and use of a data element.

- Value Domains are the included, excluded, discrete or range sets of values that are allowed within the overall context of the value meanings specified within conceptual value domains. Value domains are then either used by data elements or subsets are employed by attributes of entities, columns of tables, or DBMS columns of DBMS tables.

- Value Domain Structure represent a collection of data element concepts that are related and are in turn represented by a collective data element concept.

- Value Domain Structure Type represent the classification of structure among a set of data element concepts.

- Value Domain Data Types is a classification of the values represented by a set of data values. The data types common represented are character, integer, binary, and the like. Each data type imposes a set of rules regarding allowable values and allowed operations on the values. For example adding an integer value to a date value, but disallowing the adding of two dates.

- Value Domain Values represent the actual values that are defined within the context of a value domain.

- Value Domain Value Structures represent the mapping between values of different value domains.

- Value Domain Value Structure types represent the hierarchical classification of the mapping of value domain values. For example, Gender Value Mappings then Male Gender Value Mappings and Female Gender Value Mappings then the actual male gender values that are mapped and the female gender values that are mapped.

4 Reference Data

The only reference data table is business domain. The purpose of business domain to enable data elements that have the same fundamental name to be disambiguated. Readers are encouraged to thoroughly review and understand the Data Modeler Architecture and Concept of Operations book that is available from the Whitemarsh website, www.wiscorp.com.

5 Operation

Once the application is installed it is ready to use. Just invoke the software from the metabase program. The application is a traditional windows application. Metabase reports are accomplished through any ODBC class report writer such as Crystal Reports.

Figure 2 shows that when the metabase application is started and the Data Modeler icon is pressed, the subordinate menu for all the data modeler components is presented. When the Data Element button is pressed, the data element program is started.

5.1 Log In Process

Figure 2 shows the log-in screen that appears immediately after the application is started. Entered is your user name and your password. These are created by the Metabase Administrator

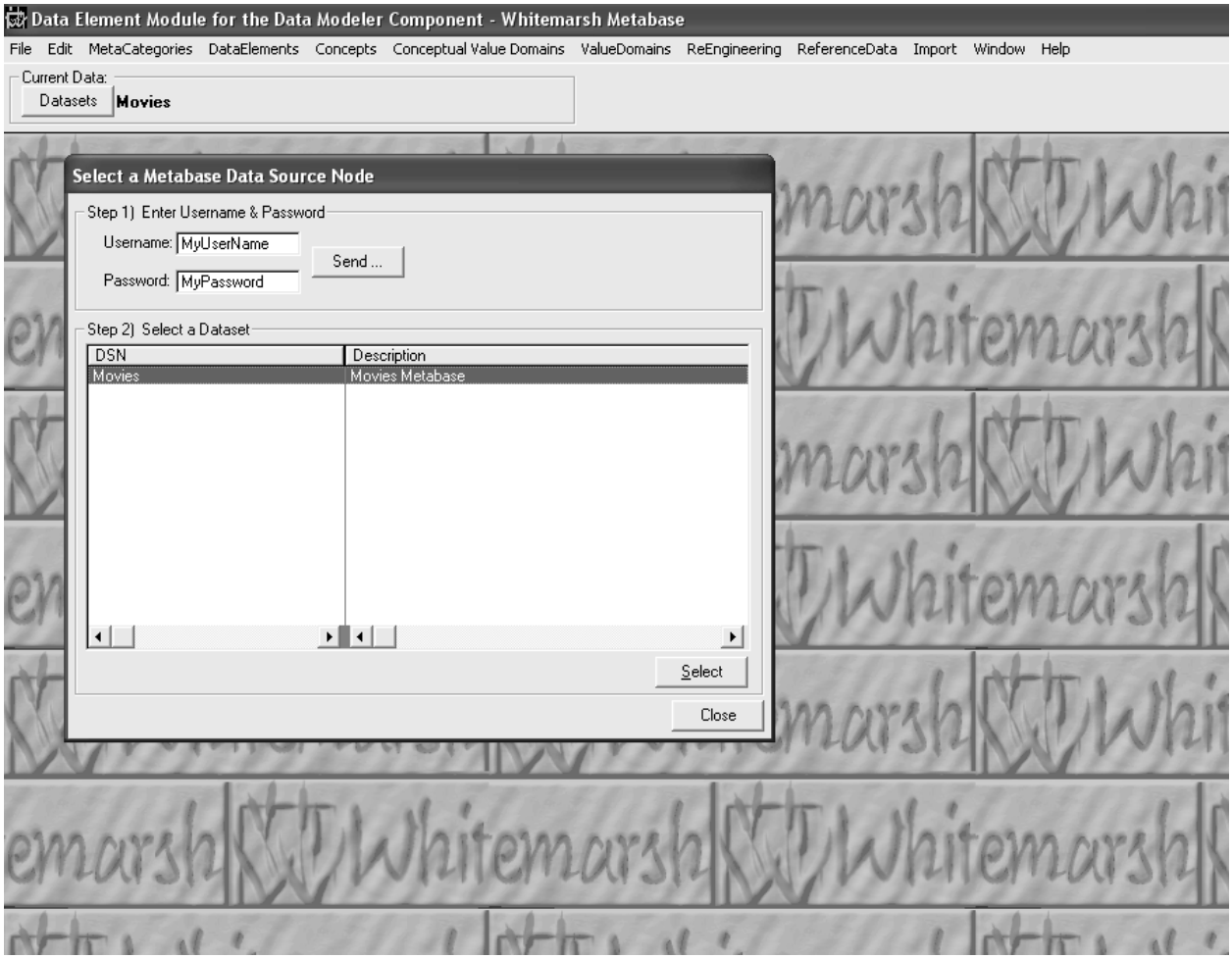

Figure 2. Login Screen.

through the metabase administration module. Please contact your metabase administrator to set up your user name and password. Once a user name and password is established, all the user's information can be changed by the user through a restricted use version of the administrator software. Once the send button is pressed the specific metabase database instances that can be accessed by the user is presented. The metabase is such that users are allowed to use specific metabase instances and specific metabase modules.

In this particular example, the user, once they sent their user name and password are shown the metabase database that they can access, that is, Movies. Highlight the choice and press the Select button. Once that is done then the metabase name, Movies, is shown as the data set that is being accessed.

6 Process Model

The data elements process consists of three classes of processes:

- Reference Data
- Fact Data Entry
- Reports

The top level menu for data elements contains the following top level items:

- Data Semantics
- Meta Categories
- Conceptual Value Domains
- Reference Data
- Export and Import
- Reports

Each menu item contains as appropriate, nested subordinate menu items. The complete menu is provided in the table that follows. When a actual process is activated, its existing list is presented. To add, change or delete an item on the browse list, the Insert, Change, or Delete button is pressed. The form that is then presented supports the entry of all the data that is needed.

Required reference data is presented first. Thereafter, the processes are from left to right and top-down according to the application's menu.

-- MetaCategories
    -- Meta Category Value
    -- Meta Category Value Types
    -- Meta Category Value Type Class

-- Data Elements

-- DE Data Hierarchy
-- Data Elements
-- Data Element Value Domains
-- Assign Data Elements to Meta Category Values

-- Data Element Classifications
-- Data Element Classification
-- Data Element Classification Structures
-- Data Element Classification Structure Types
-- Assign Data Element Classifications to Data Elements

-- Compound Data Elements
-- Compound Data Elements
-- Compound Data Element Structures
-- Compound Data Element Structure Types
-- Assign Compound Data Elements to Data Elements

-- Derived Data
-- Derived Data Elements
-- Assign Derived Data Elements to Data Elements
-- Assign Derived Data Elements to Compound Data Elements

-- Data Element Concepts
-- Data Element Concept Data Hierarchies
-- Data Element Concepts and Data Elements
-- Data Element Concept
-- Data Element Concept Structures
-- Data Element Concept Structure Types
-- Assign Data Element Concepts to Meta Category Values

-- Concepts
-- Concepts
-- Concept Structures
-- Concept Structure Types
-- Concepts, Data Element Concepts and Data Elements

-- Conceptual Value Domains
-- Conceptual Value Domain
-- Conceptual Value Domain Structures
-- Conceptual Value Domain Structure Types
-- Conceptual Value Domains, Data Element Concepts and Data Elements

-- Value Domains

-- Value Domains
-- Value Domain Structures
-- Value Domain Structure Types
-- Value Domain Values
-- Value Domain Value Structures
-- Value Domain Value Structure Types

-- ReEngineering
-- Reassign Data Elements to Business Domains
-- Reassign Data Elements to Data Element Concepts
-- Reassign Data Elements to Value Domains
-- Reassign Data Element Concepts to Concepts
-- Reassign Data Element Concepts to Conceptual Value Domains
-- Reassign Value Domains to Conceptual Value Domains
-- Reassign Value Domain Values to Value Domains
-- Promote Data Element to Data Element Concept
-- Promote Data Element Concept to Concept
-- Remove Data Element Concept to Meta Category Value Assignments
-- Remove Data Element to Meta Category Value Assignments
-- Reallocate Business Domains
-- Reallocate Meta Category Values
-- Reallocate Meta Category Value Types
-- Reallocate Meta Category Value to Different Type

-- Reference Data
-- Business Domains
-- Value Domain Data Types

-- Import
-- Import Data Element Concepts
-- Import Conceptual Value Domains
-- Import Concepts
-- Import Data Elements
-- Import Data Element Classifications
-- Import Value Domain
-- Import Value Domain Values

Menu for Data Elements

6.1 Reference Data

Business Domains serve to disambiguate data elements that may otherwise have the same name. This is needed because many terms, such as track have different meanings within different disciplines. Figure 3 presents the current set of business domain reference data. Each business domain is listed along with the data elements associated with that domain.

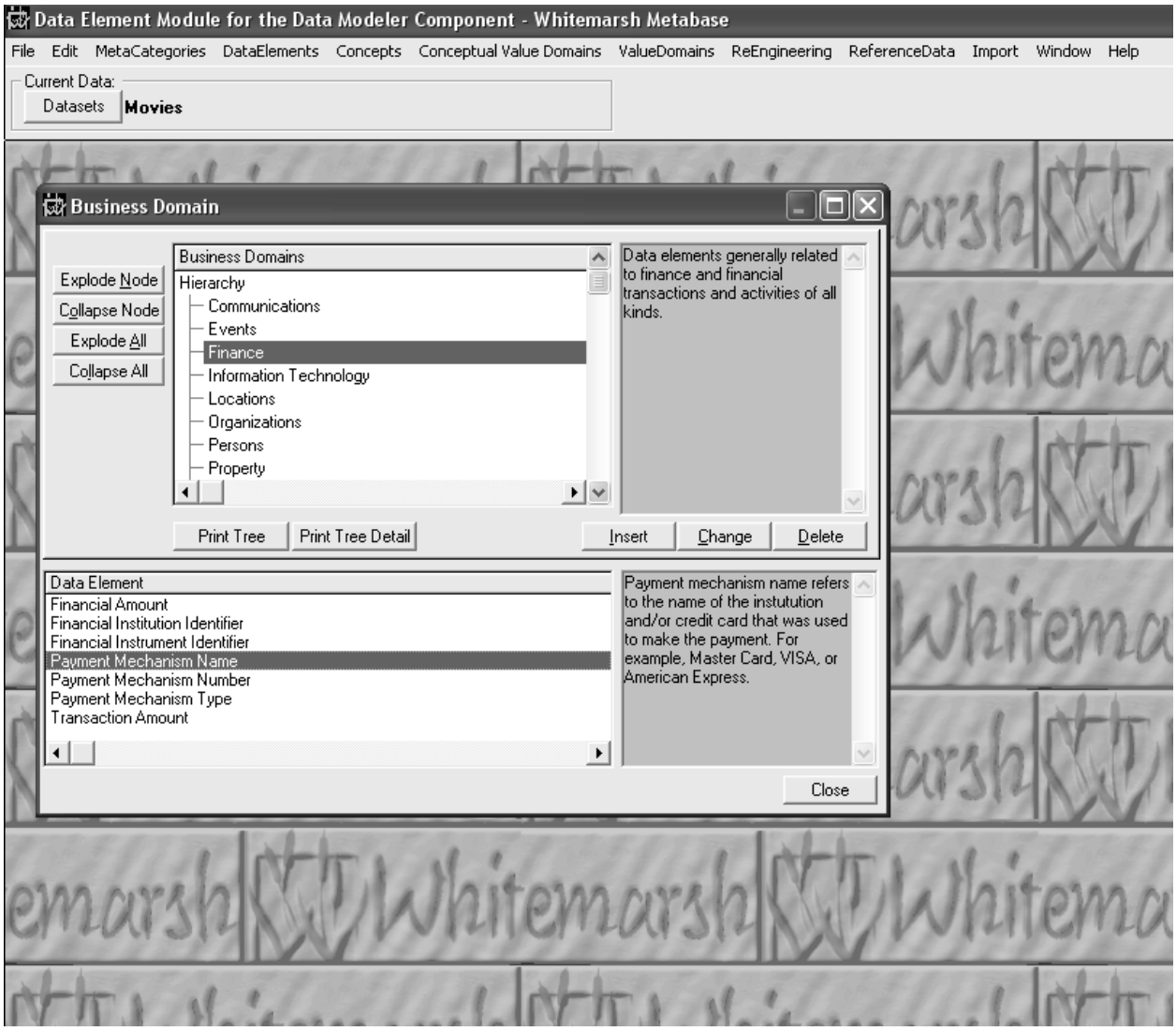

Figure 3. Business Domain list.

To add, change or delete from this list, the Insert, Change or Delete buttons are pressed. Referential integrity in the metabase is conservative. That is, On Delete: Restrict. That means that if a row of data that is to be deleted is employed any where else in the metabase, the deletion is refused

Figure 4 presents the update form for entering or changing a business domain. The only data that is required is the name of the business domain and its description. The description should be sufficient to then disambiguate any two or more data elements that then have the same name. The business domain's description becomes part of the data element's complete semantics.

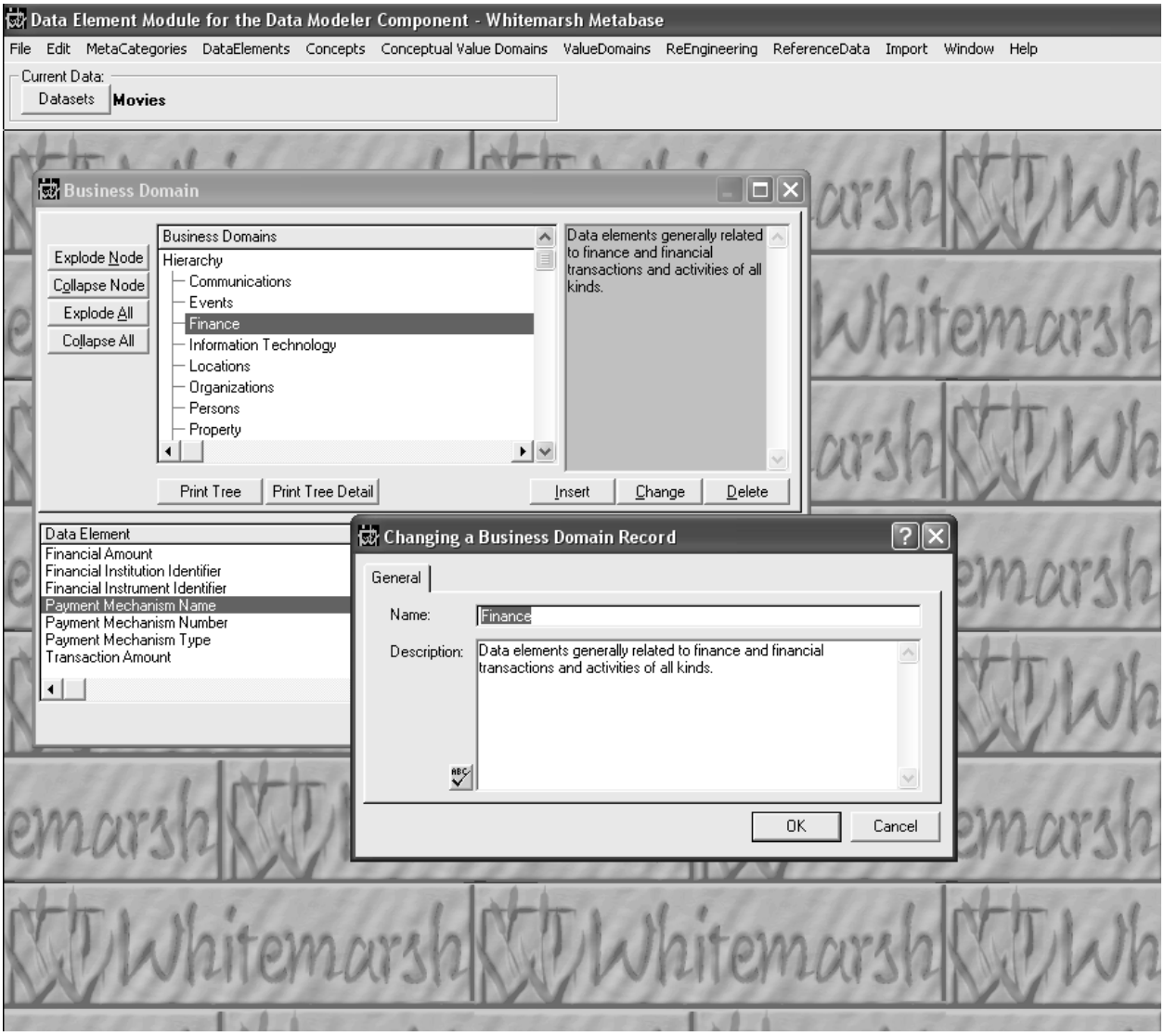

Figure 4. Business Domain update screen.

6.2 Fact Data

The fact data is broadly defined into the following categories:

- Data Semantics
- Meta Categories
- Concepts
- Conceptual value domains, and
- Reference Data

6.2.1 Data Semantics

Within Data Semantics is:

- Data Elements
- Compound Data
- Derived Data, and
- Reverse Engineering

6.2.1.1 Data Elements

A full exposition of data elements includes:

- Data element data hierarchies
- Data elements
- Data element value domains
- Assignment of meta category value

6.2.1.1.1 Data Element Data Hierarchies

The objective of the data element component of the data modeler module of the Whitemarsh metabase is the development of context independent business fact template that can be employed to semantically define attributes of entities, columns of tables, and the semantics of DBMS columns in operational databases. Figure 5 illustrates that objective very clearly. Event date, a data element is employed a number of times in the Specified Data Model attribute, which in turn is employed two times in an Implemented Data Model column, which finally, is employed as the semantics for one Operational Data Model DBMS column. This is a clear example of the "define once and use many time principal of the metabase.

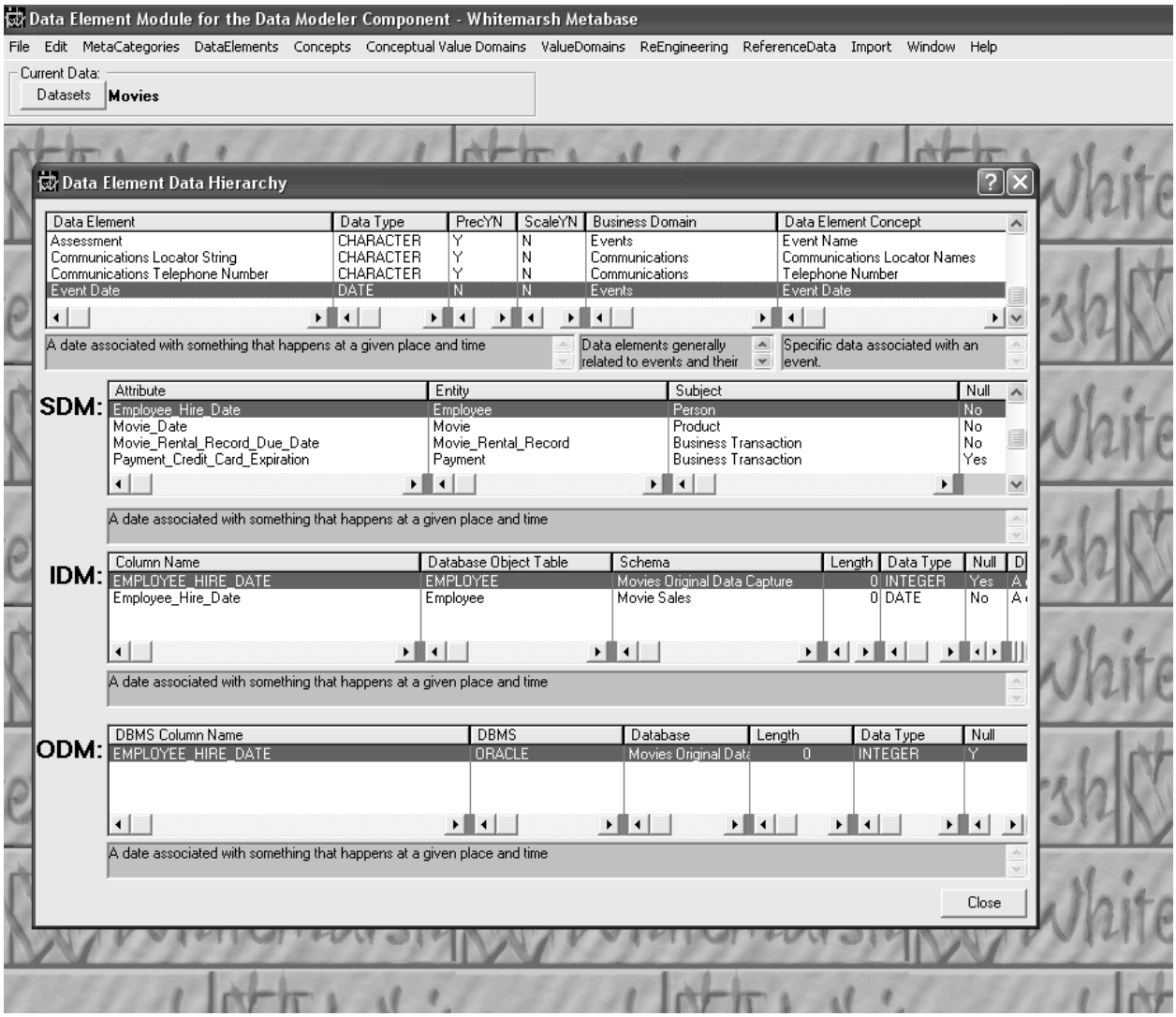

Figure 5. Data Element hierarchies showing data elements across all data model levels.

6.2.1.1.2 Data Elements

Figure 6 presents a current set of data elements within business domains. For each data element the assigned meta categories also appear. To add, delete, or change a data element, first select the appropriate business domain, and then press the appropriate buttons. Data elements can only be deleted if they are not otherwise employed. Shown also in this figure are the meta category values that have been assigned. Finally shown are two forms of definition for the data element: the local definition, and the contextual definition. Both forms of definitions are created on the Data Element update screen.

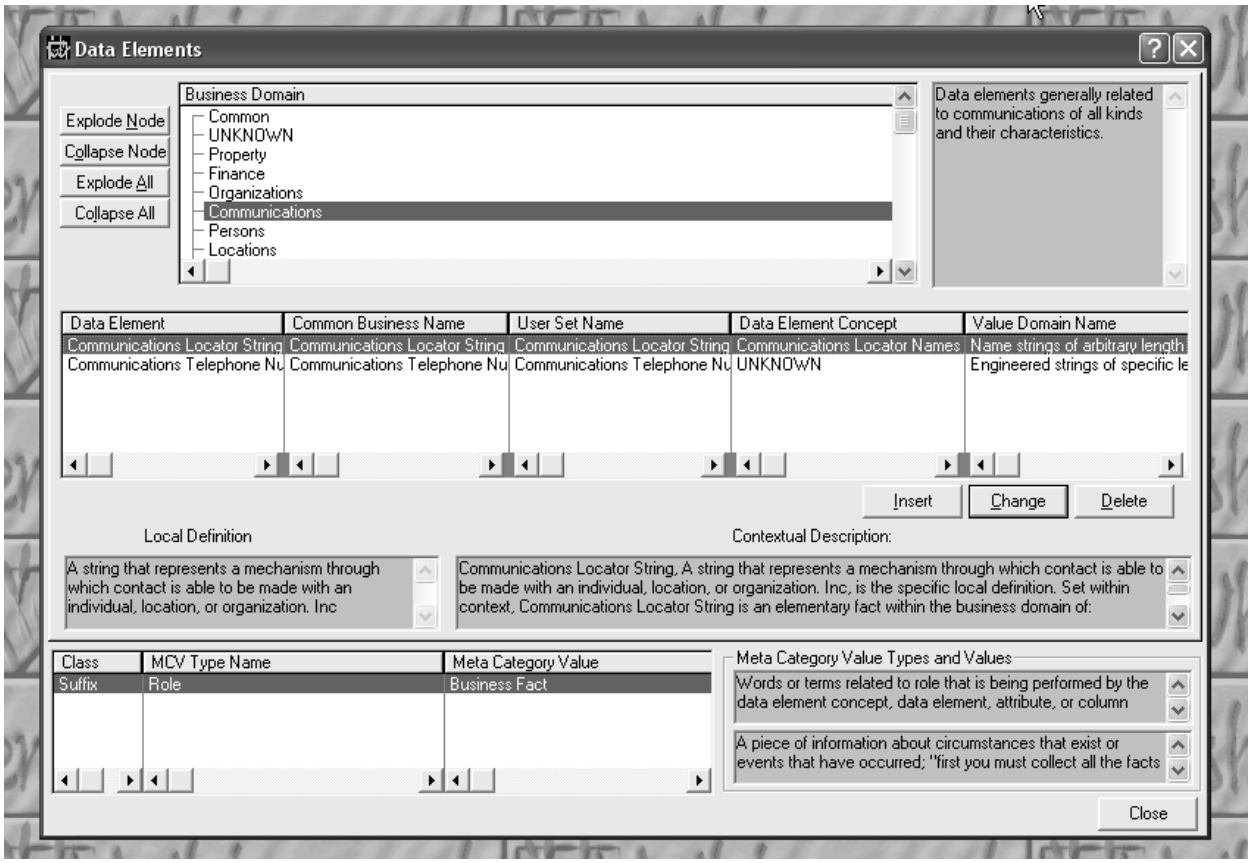

Figure 6. List of data elements within a business domain.

Figure 7 presents the data entry form for a data element. A data element exists within the context of a business domain, data element concept, and value domain. It has a common business name, three standard abbreviations, a automatically generated data element name, and a user set name. Finally it has a description.

When a data element is changed the check box, freeze all names, is automatically checked. This is to prevent against accidental name changes. If the box is unchecked then components of the data element's name can be changed. These are: common business name, abbreviations and user set name. All the semantics assigned or inherited remain connected. That is, business domain, data element concept, and assigned meta category values. This ensures that even though a data element's name may be changed, its semantics remain.

When a new data element is entered, the business domain has already been selected.

Every data element has two definitions: local, and in context. The local definition should be very short, just a simple phrase. Do not make this a sentence. Do not start it with a capital letter, nor end it with a period.. The contextual definition is automatically created through processes activated by the AutoDef button. At the end of the AutoDef process, the local definition is included within the first sentence.

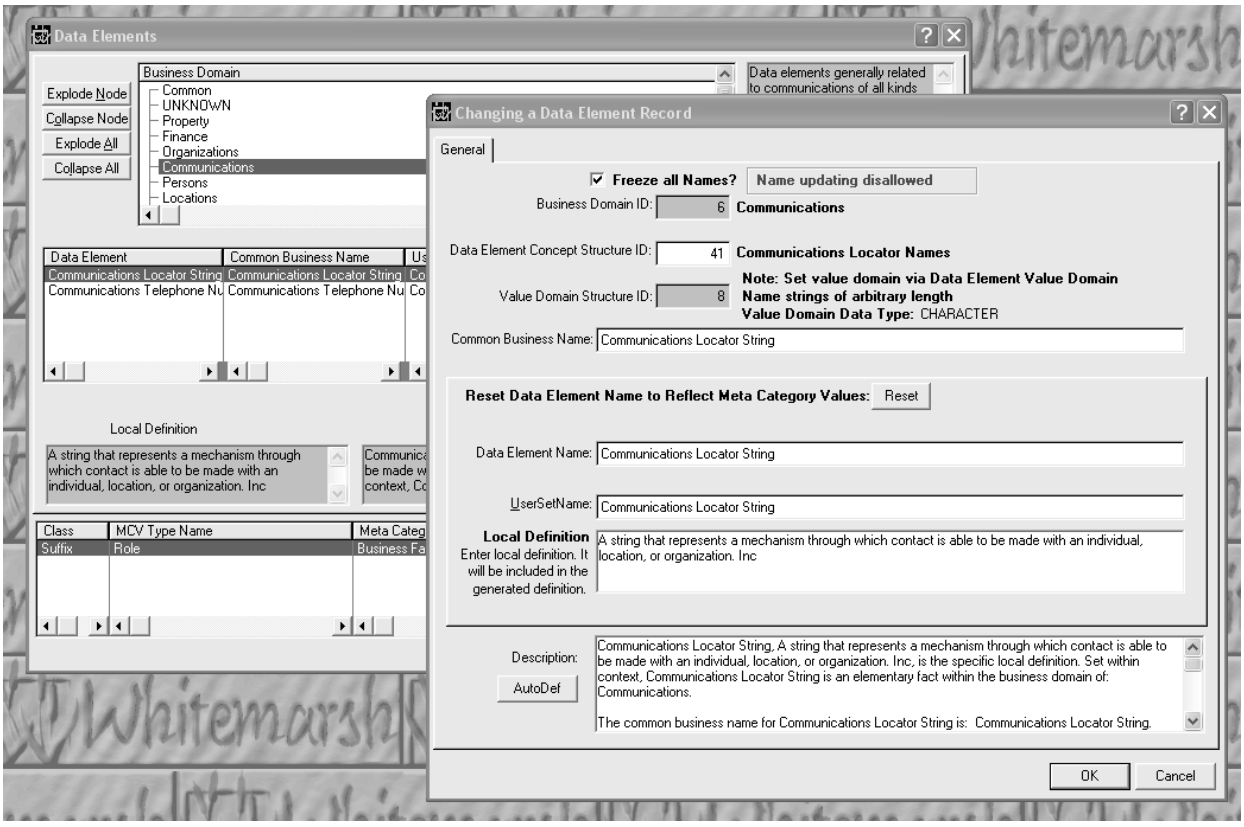

Figure 7. Data element update screen.

The next step is to select the data element's data element concept. Figure 8 presents a list of data element concepts. This list appears when the value for a data element's data element concept is zero and then the field is tabbed through. Again, select the appropriate one and press the select button. This is accomplished by highlighting the appropriate concept type, then concept, then conceptual value domain structure type, then conceptual value domain, then the appropriate data element concept structure type, and then finally, the appropriate data element concept within its data element concept structure hierarchy. At that point, press the Select button.

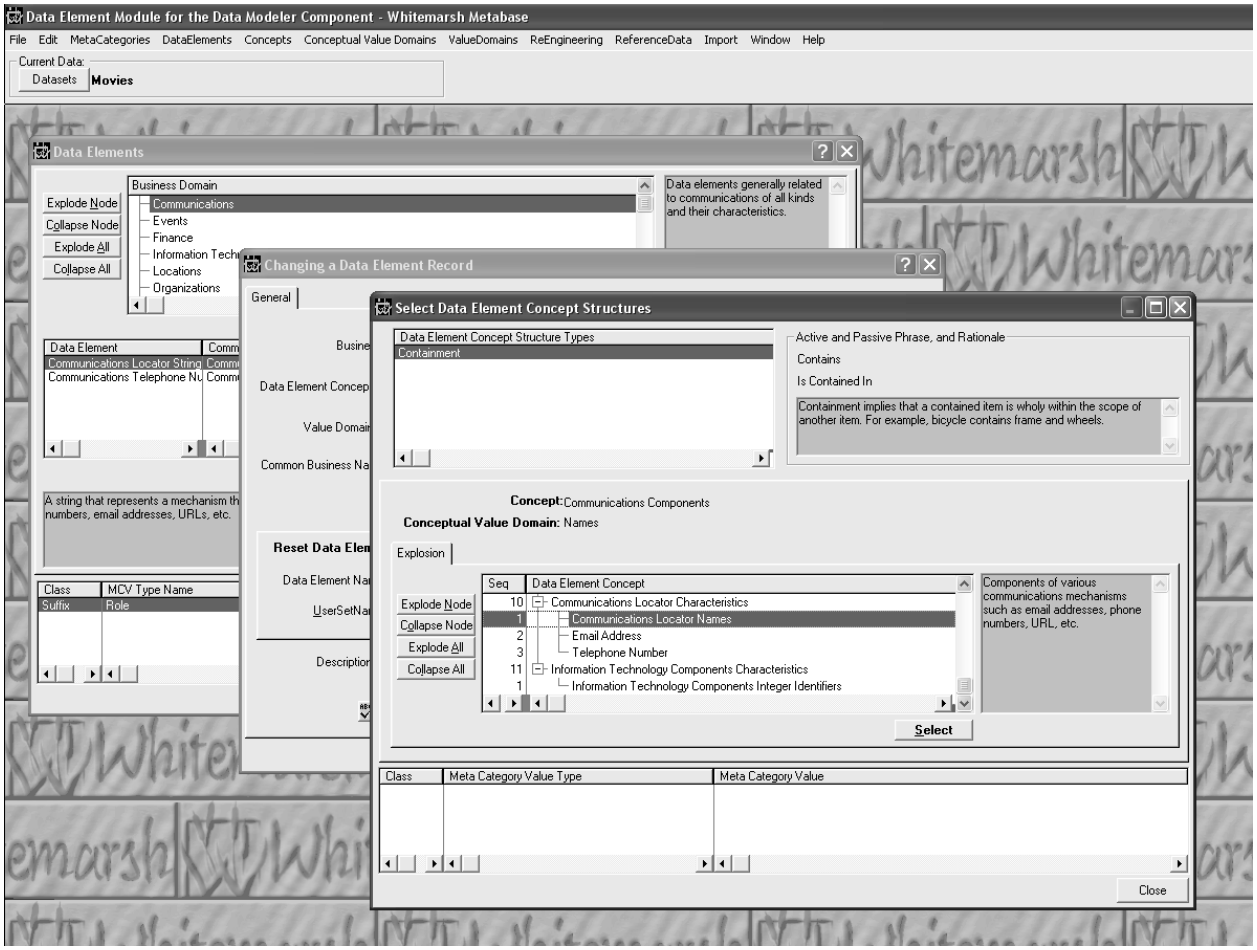

Figure 8. Data Element Concept selection screen.

The next step is to select the appropriate value domain that is to govern the value set for the data element. When a new data element is created it's value domain is set to 1, that is, Unknown. Value domains can only be assigned to data elements through the Data Element Value Domains menu item.

When all three are selected, the data element's common business name is automatically created as the combination of the <business domain name> and <data element concept> phrases. This name can be changed. Even though it is changed, the relationships back to the business domain and the data element concept remain and then serve as semantics for the data element.

A data element may also have three sets of abbreviations. Abbreviations are not aliases. If a data element is enhanced through the allocation of additional semantics, that is, the assignment of meta category values (see a section below for just this purpse), the data element's name must change to reflect their addition. The name change is automatically accomplished by pressing the Reset button. All the assigned prefix and suffix meta category words are attached to the beginning or the end of the data element's common business name.

6.2.1.1.3 Data Element Value Domains

Data element value domains represent the assignment of a value domain to a data element. The assigned value domain must be within the same conceptual value domain as the data element's data element conceptual value domain. The conceptual value domain browse window contains both data element concepts and value domains.

Figure 9 presents the data element value domain assignment window. To arrive at the set of data element value domains, proceed left to right and top-down. Highlight and select the concept structure type, then concept. Then select a data element concept structure type, then data element concept. That will then list the data elements. Then select a value domain structure type and then value domain. At that point, the data elements and their value domains are within the context of concepts, data element concepts, and value domains appear at the bottom browse.

To then assign a particular value domain to a particular data element, highlight the appropriate data element and the appropriate value domain, then press the Assign button (lower right). If an attribute, column or DBMS column is highlighted then the Assign and Set to Unknown buttons are disabled.

If the value domain is allowed to be assigned then a message to that effect is displayed in the message section. A value domain is allowed to be assigned if it is not already assigned to a contained attribute, column, or DBMS column.

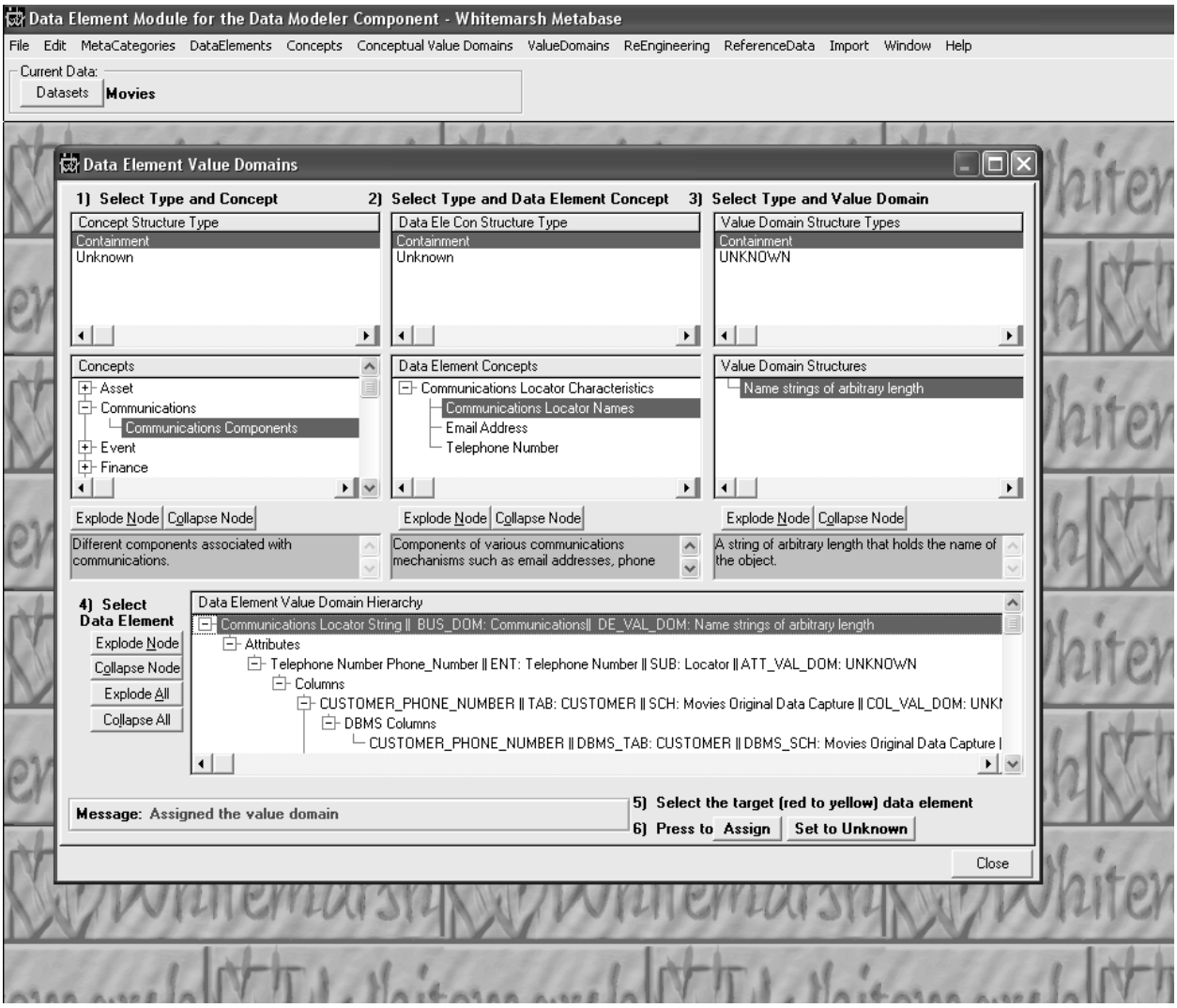

Figure 9. Data Element Value Domains.

6.2.1.1.4 Data Element Meta Category Values

The semantics of data elements are both inherited and assigned. They are inherited through the selected business domain and the data element concept. Assigning semantics is illustrated in Figure 10. In this assignment process, the appropriate business domain is highlighted. Then the Data element is highlighted. It is then tagged by either checking in the first column or pressing the tag button. If more than one is tagged, only the first is "remembered." If there are any already assigned semantics then they are shown in the bottom browse window.

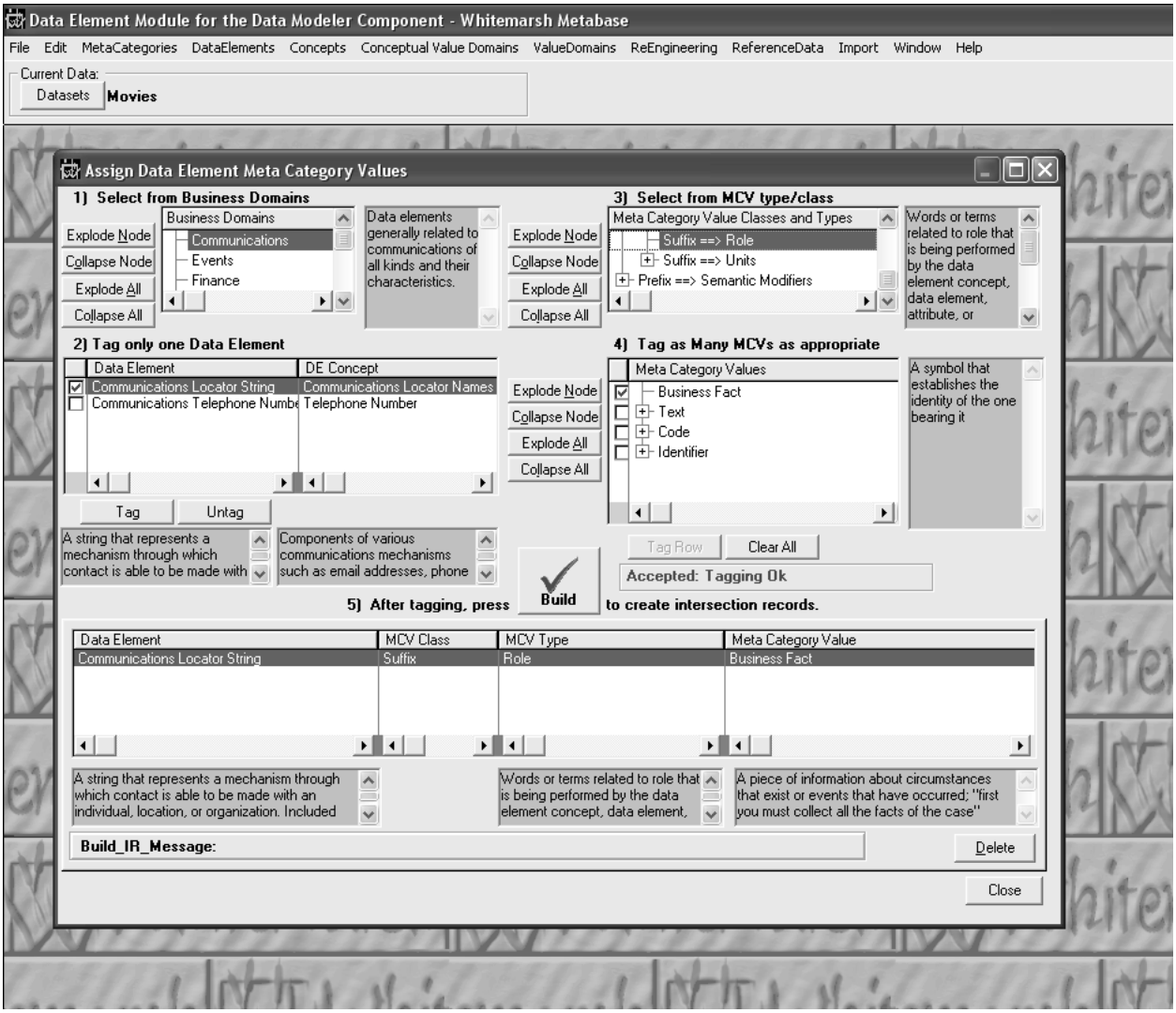

Figure 10. Data Element Meta Category Value assignment.

The right two windows contain the set of meta category value types and type classes, and the meta category values for a particular highlighted meta category value. Only one meta

category value from each meta category value type can be tagged. For example, it makes no sense to assign both estimated and projected to the same data element.

Suffix meta category values will then appear at after the data element's common business name. Prefix meta category values will appear prior to the data element's common business name. The order of appearance is not arbitrary or able to be change by a metabase end-user. The order is specified within the table, meta category type. So, if the geographic meta category value type's sequence number is 2 and the temporal meta category value type's is 3 then regardless of the sequence of tagging, geographic meta category values will always be prefixed into the data element's constructed name before those for the temporal meta category value type.

Once the meta category values are tagged and the build button is pressed, the meta category values are assigned to the data element. The assignment process determines one by one that the tagged meta category values are semantic subsets of those already assigned either to a "parent" data element concept, or to a "child" attribute or column. If the tagged semantic is a semantically acceptable then it is assigned. Otherwise the assignment for the offending semantic is rejected.

Once a semantic is assigned, if the data element is employed in the creation of an attribute or column, or if a column that is a deployment of a data element is employed in the creation of a DBMS column, then none of the semantics for the data element can be deleted. New ones can be added, but none can be deleted.

When ever semantics are added, the data element's name remains the same until it is reset in the data element update screen.

6.2.1.2 Data Element Classifications

A data element classification is a scheme through which data elements are classified. Data element classifications consist of:

- Data element classification
- Data element classification structures
- Data element classification structure types
- Data element to data element classification assignments

6.2.1.2.1 Data Element Classification

A data element classification is a mechanism through which one or more data elements is classified. Data element classifications, shown in Figure 11, are independently created and then assigned to one or more data elements. Figure 12 presents the data update form for adding, deleting or changing a data element classification.

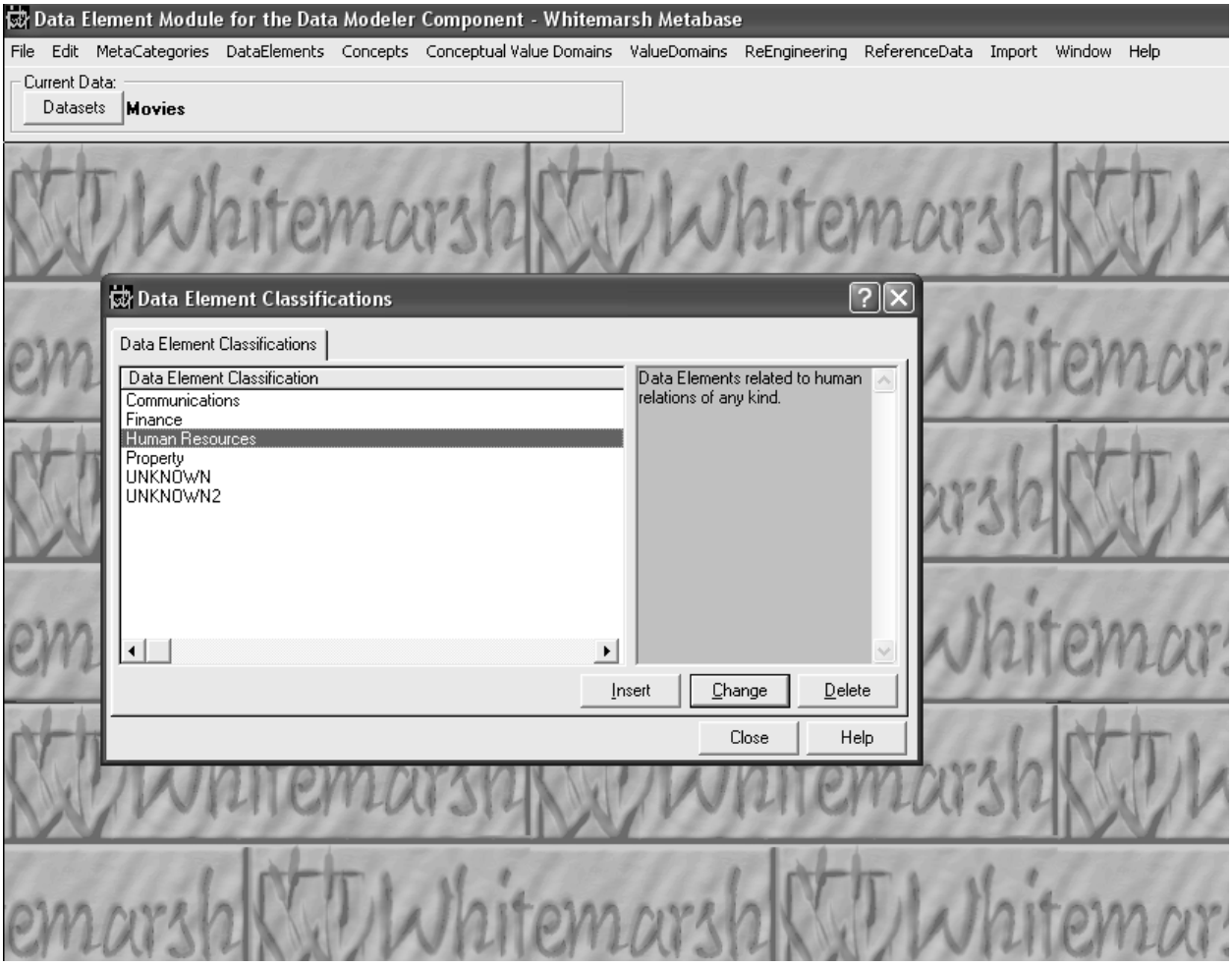

Figure 11. Data Element Classifications.

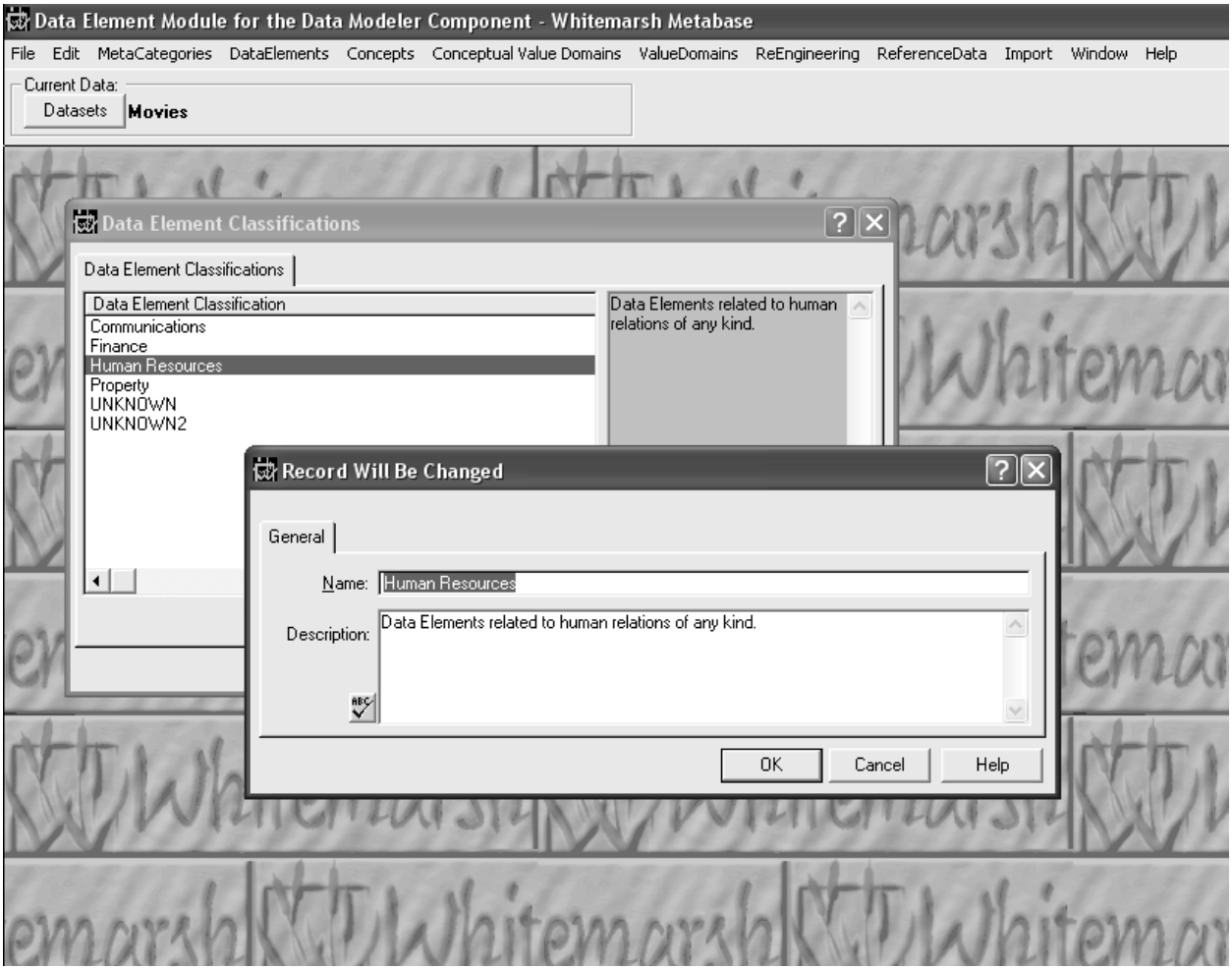

Figure 12. Data Element Classification update screen.

6.2.1.2.2 Data Element Classification Structures

Data element classifications can exist singly or in hierarchies, or networks. In the last case, data element classifications form a traditional bill of materials data structure. Users are encouraged to view the Whitemarsh application WisBOM, Whitemarsh Bill of Materials. This application can be downloaded from the Whitemarsh website and it contains it's own data examples, supporting diagrams and documentation. This application is provided as a training device.

Figure 13 shows a existing set of data element classification structures. In this example, the same data element classification is contained in two other data element classifications. Hence, the bill of materials.

When a data element classification name is highlighted and the Insert button is pressed, a screen like Figure 14 is presented.

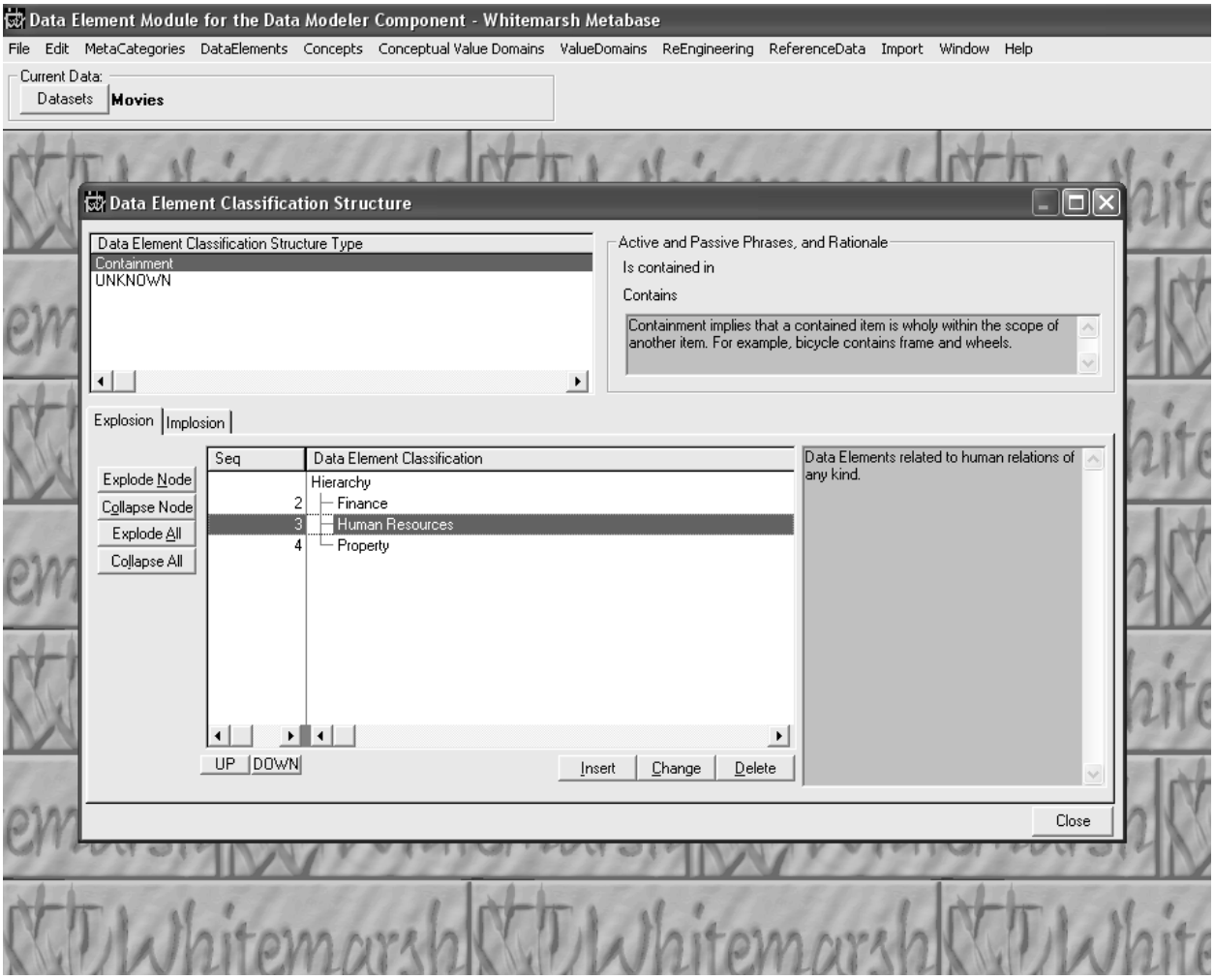

Figure 13. Data Element Classification Structures.

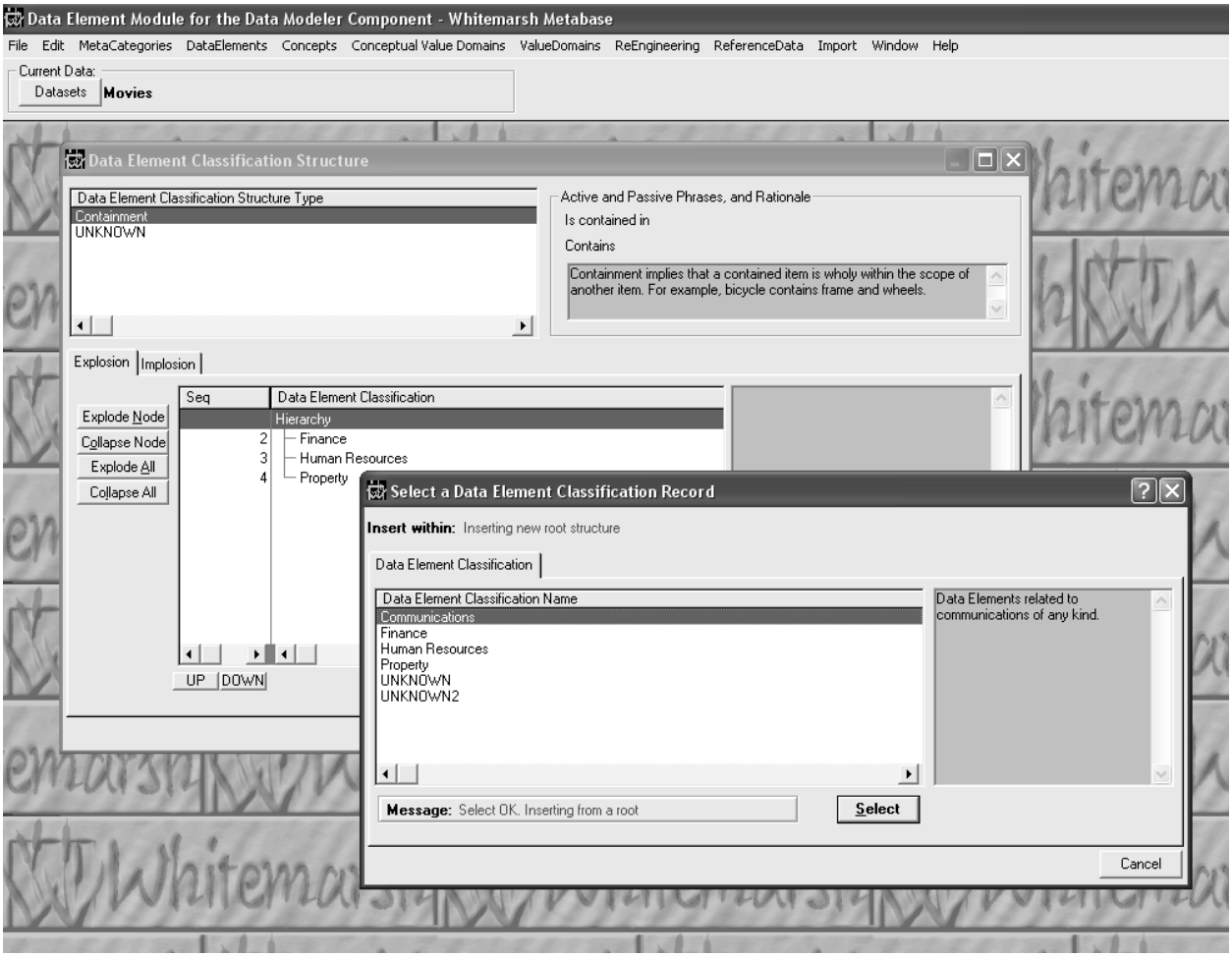

Figure 14. Selecting a Data Element Classification into a Data Element Classification Structure.

The specific data element classification that is desired as the contained data element classification is highlighted and then the select button is pressed.

There are three cases that are automatically prevented: twins, recursion and infinite recursion. A twin the case would be inserting an additional Finance. This is what the select in this figure is attempting. The Select button is gray, that is, disabled. The figure additionally shows the message that identifies the reason why the Select button is disabled.

The second screened-out case is the insertion of a Finance as a child of the existing Finance, that is, inserting the same as selected. The third case would be inserting Finance after inserting Communications within Finance. This would have the effect of establishing an infinite recursion.

These three cases are automatically screened out in all Bills-of-Materials data structures within the metabase. In the data element module there are BOMs for conceptual value domain, value domain, data element concepts, data element classifications, and compound data element.

6.2.1.2.3 Data Element Classification Structure Types

The data element classification structure type is a way of distinguishing one classification structure from another. Figure 15 presents the current list. If two different classification hierarchies are interconnected, it may be that the intersection is distinguished from the others by means of a different data element classification structure type.

Figure 16 presents the data element classification structure type update form. Not only is the name and description of the data element classification structure type provided, so too is the active phrase and the passive phrase. The active phrase is employed by the Whitemarsh metabase system when a down-ward structure is presented. For example, <parent> contains <child 1>, <child 2>, ..., <child n>. The passive phrase enables the reverse phrases to be presented. That is, <child 2> is contained in <parent>.

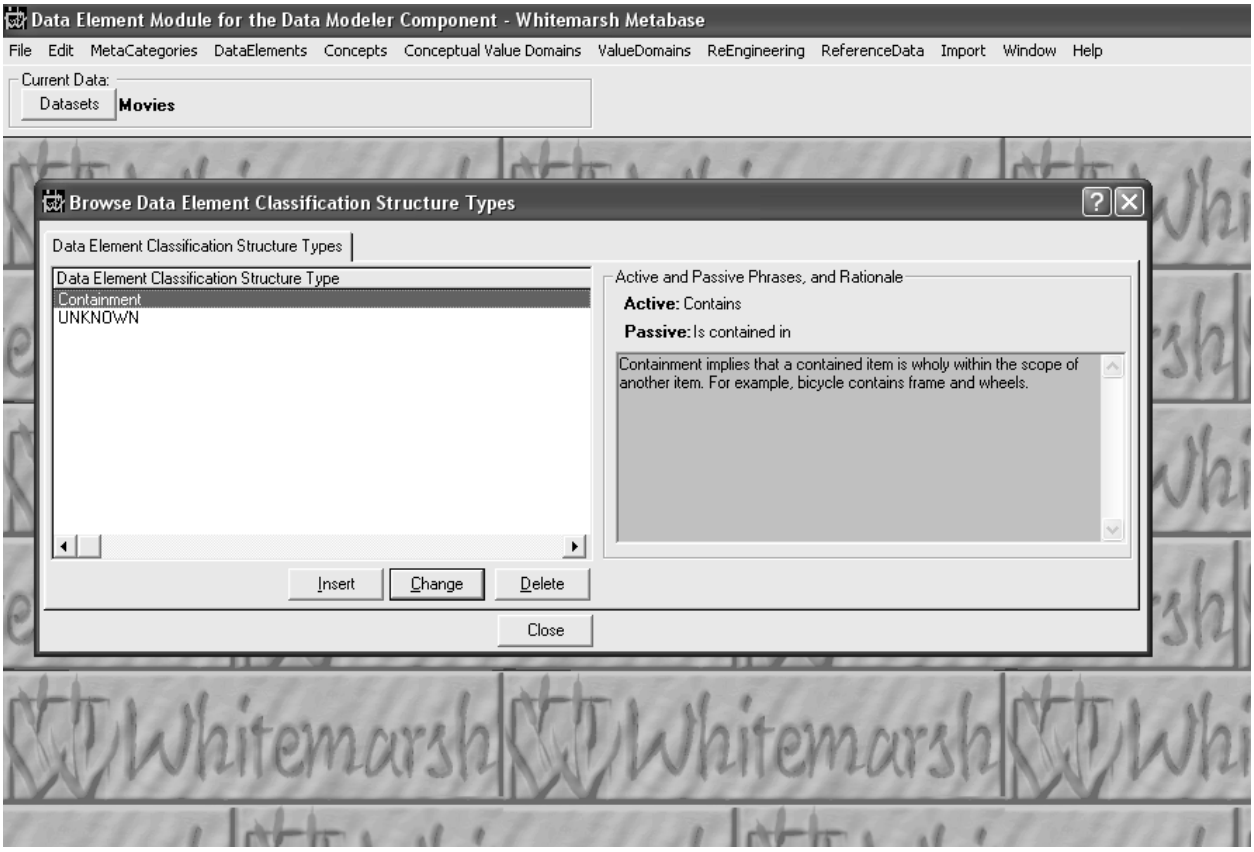

Figure 15. Data Element Classification Structure Types.

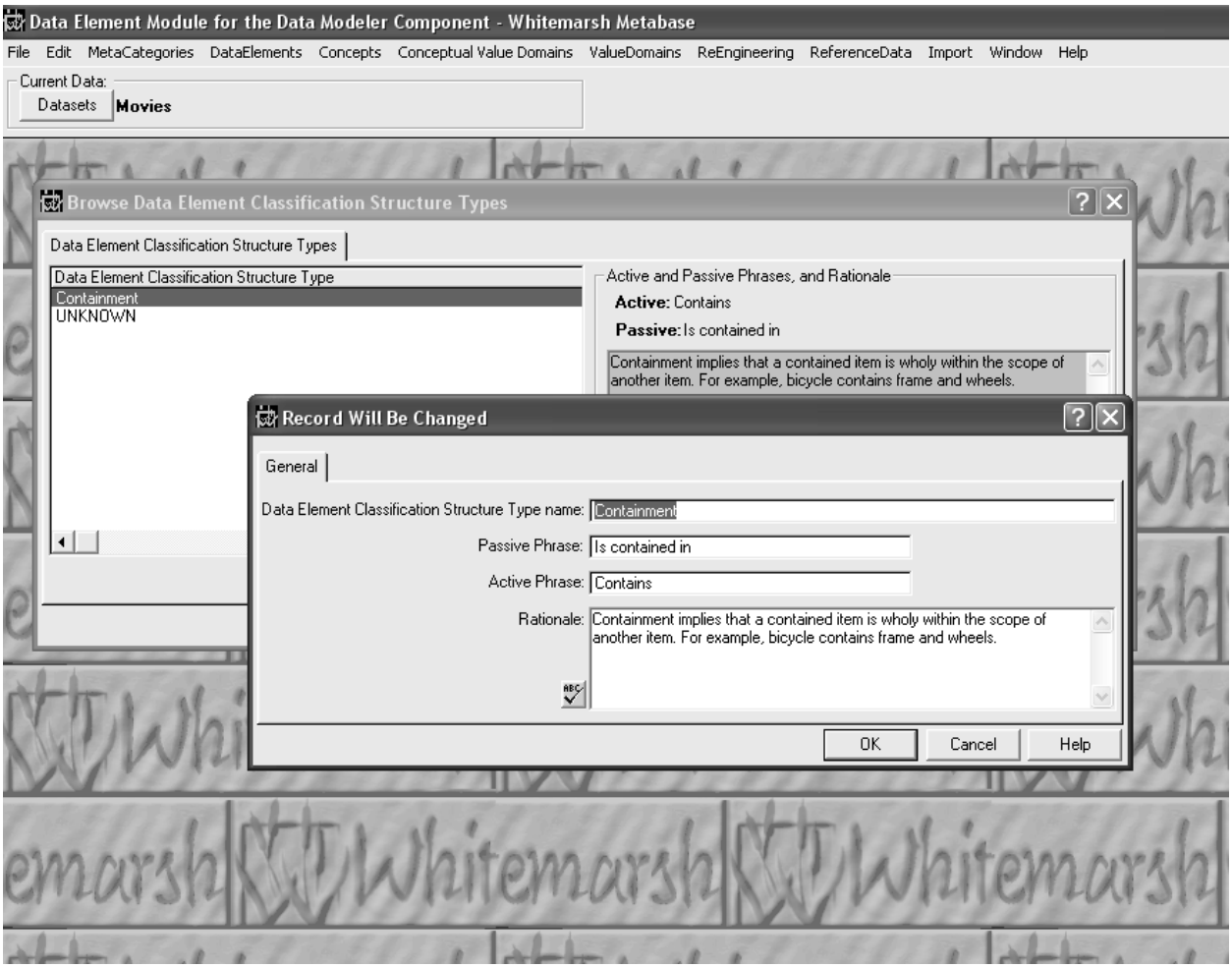

Figure 16. Data Element Classification Structure Type update screen.

6.2.1.2.4 Assignment of Data Element Classifications to Data Elements

A data element may participate in one or more classifications. Figure 17 presents the window that shows classifications already assigned to a data element. This same window permits new classifications to be established, and finally, to delete existing classifications.

To assign a data element to an already created classification, highlight the data element in the upper left window. If more than on is tagged, only the last one tagged is employed. Tag that data element via the tag button or by mouse-clicking in the check-box to the left of the data element. Then, highlight one or more classifications in the upper-right window. Tag or mouse-

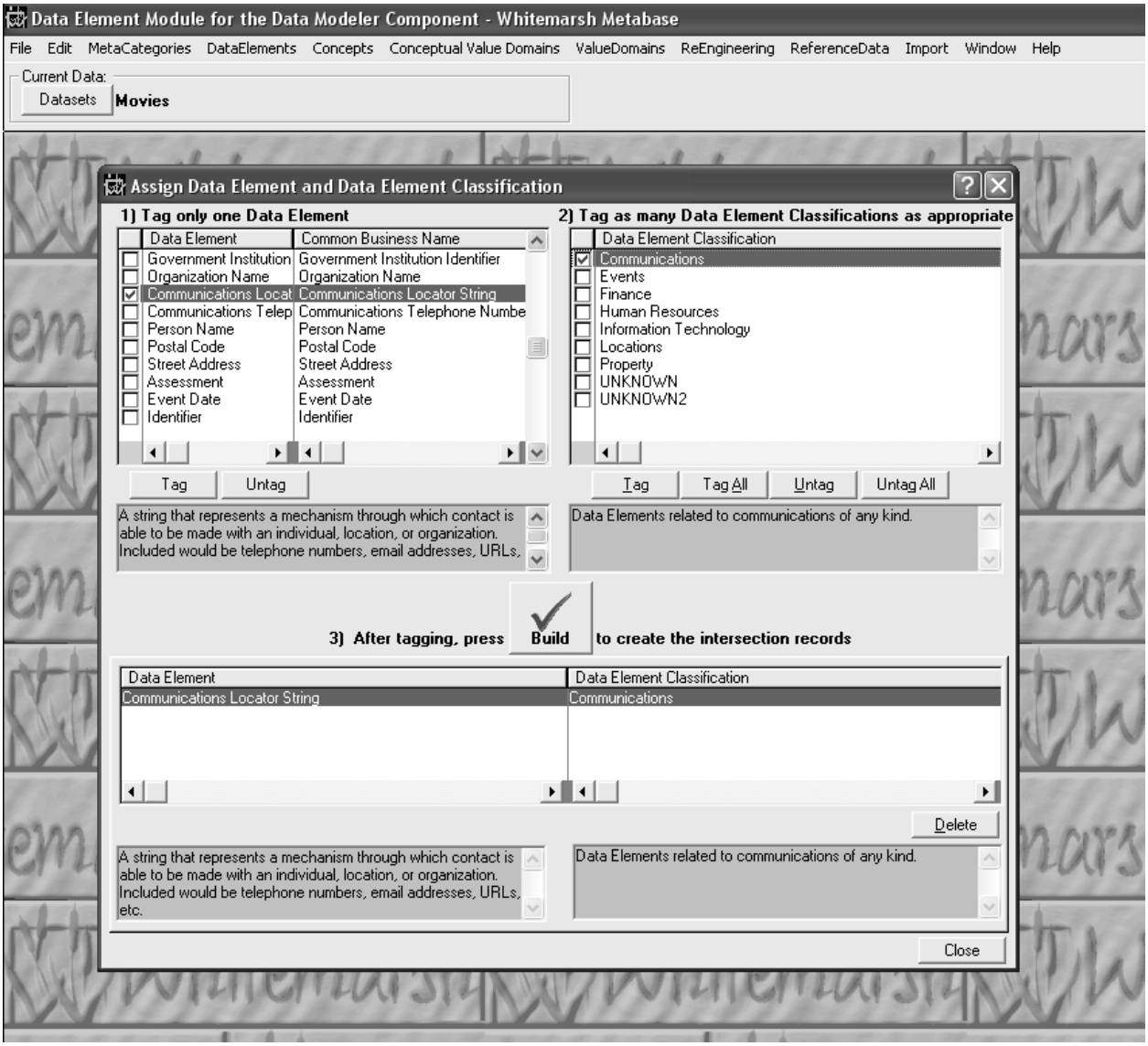

Figure 17. Data Element to Data Element Classification assignment.

click each one. Once all have been tagged, then press the Build button. Each relationship

between the one tagged data element and its assigned data element classifications is created and then displayed in the bottom window.

6.2.1.3 Data Element Concepts

Data element concepts are the set of concepts that support the understanding of a data element. Data element concepts exist within conceptual value domains (see Section 6.2.4) and concepts (see section 6.2.3). A data element concept is a broad grouping of data elements within a conceptual value domain. A conceptual value domain is employed to broadly define the underlying concepts within which the data element concepts reside. For example, a conceptual value domain might be business organization components. Within that, a contained data element concept might relate to addresses, corporate identifications, finance, and the like. Then, within a specific data element concept of addresses there could be subordinate address types for postal addresses, electronic addresses, and shipping addresses. Specific data elements within the data element concept, postal address, could include business name, street number, street name, city, state, country, and postal code.

Data element concepts can be complex. That is, a data element concept can be composed of multiple contained data element concepts. Additionally, a data element concept can be in multiple structures of contained data element concepts. This is a classic bill-of-materials. Included therefore are:

- Data element concept data hierarchies
- Data element concept concepts
- Data element concept structures
- Data element concept structure types
- Data element concept meta category values

6.2.1.3.1 Data Element Concept Data Hierarchies

Data element concept data hierarchies provide a cross reference between data element concepts, data elements, and the use of data elements within the specified, implemented, and operational data models. Figure 18 illustrates this cross reference.

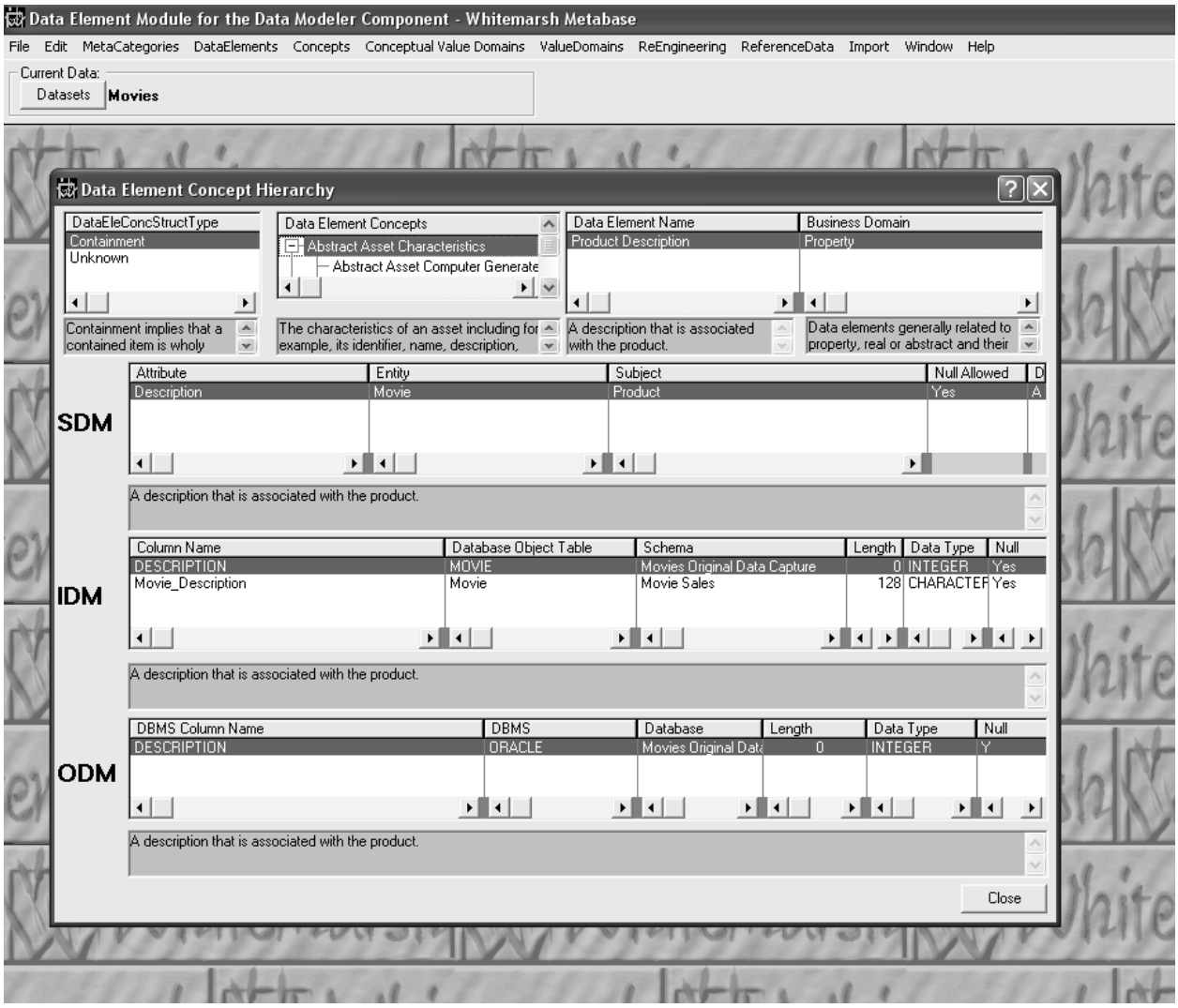

Figure 18. Data Element Concept hierarchy.

6.2.1.3.2 Data Element Concept

As stated above, a data element concept is a broad grouping of data elements within a concept and also within a conceptual value domain. For example, a concept might be business components, and a conceptual value domain associated along with that concept to then create the data element concept might be addresses, corporate identifications, finance, and the like. Figure 19 presents a list of data element concepts. The highlighted data element concept exists within the highlighted concept and also conceptual value domain. If there are any assigned meta category values assigned to the data element concept then they are listed in the bottom browse list.

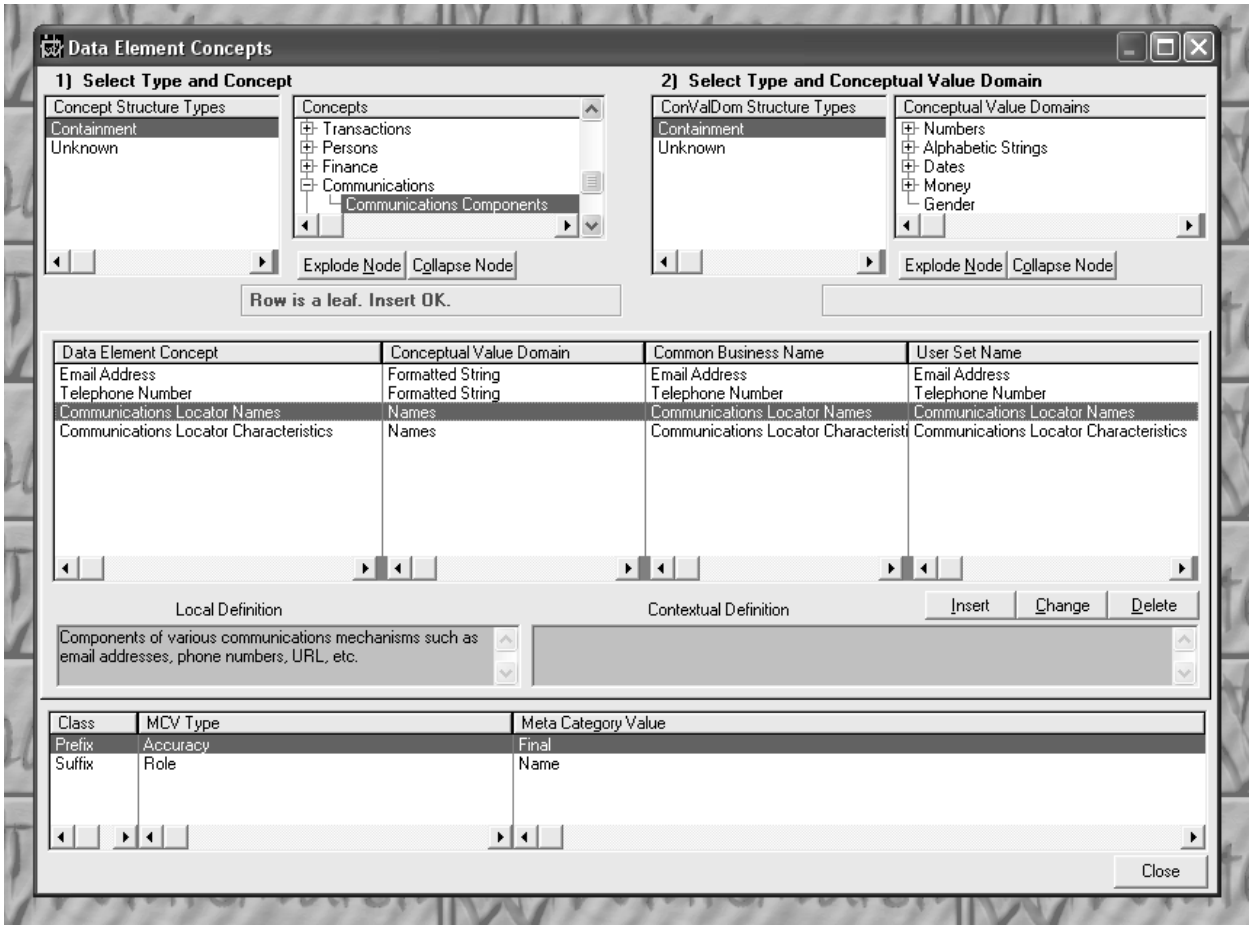

Figure 19. Data element concept.

Every data element concept can have two definitions: a local one and a contextual one. The local definition should be very short, just a simple phrase. Do not make this a sentence. Do not start it with a capital letter, nor end it with a period. The contextual definition is automatically created through processes activated by the AutoDef button. At the end of the AutoDef process, the local definition is included within the first sentence.

To add, change, or delete a data element concept press the appropriate button. Figure 20 presents the update screen.

Figure 20 does not permit the selection of a concept from within a concept structure, nor can you select a conceptual value domain from within a conceptual value domain structure. The concept and conceptual value domain is selected via Figure 20 prior to the Insert button being pressed. Changing the concept and/or the conceptual value domain of a data element concept is accomplished through the ReEngineering processes described later in this user guide.

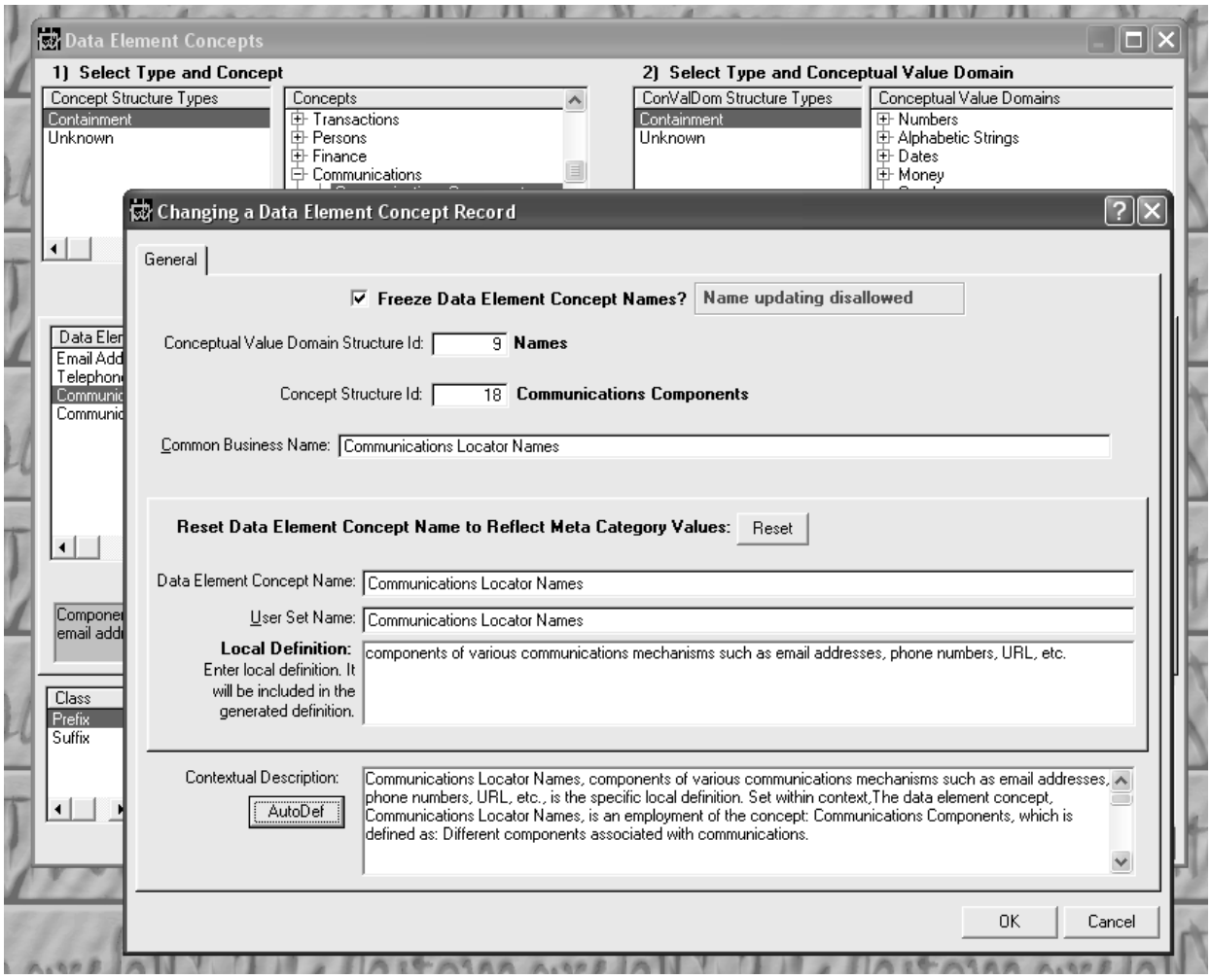

Figure 20. Data Element Concept update screen.

If the screen is for an update, the Freeze Data Element Concept Names box before the names can be changed.

The contextual definition is created by pressing the AutoDef button. If there is a local definition, it is included in the contextual definition's first sentence.

6.2.1.3.3 Data Element Concept Structures

Data element concepts can exist singly or in hierarchies, or networks. In the last case, data element concept structures form a traditional bill of materials data structure. Users are encouraged to view the Whitemarsh application WisBOM, Whitemarsh Bill of Materials. This application can be downloaded from the Whitemarsh website and it contains it's own data examples, supporting diagrams and documentation. This application is provided as a training device.

Figure 21 shows a existing set of data element concept structures. In this example, the same data element concept is contained in two other data element concepts. Hence, the bill of materials.

When a data element concept name is highlighted and the Insert button is pressed, a Figure 22, the data element structure selection screen is presented. This screen contains a listing of data element concepts that can be selected to then participate in a data element concept structure. In this particular example, the data element concept, human characteristics, has four contained concepts. When the insert button was pressed, a screen appears that enable the selection of an additional data element concept that is to be included in the existing set.

In addition to preventing twins, recursion, and infinite recursion, the metabase enforces that all data element concepts be from the same conceptual value domains.

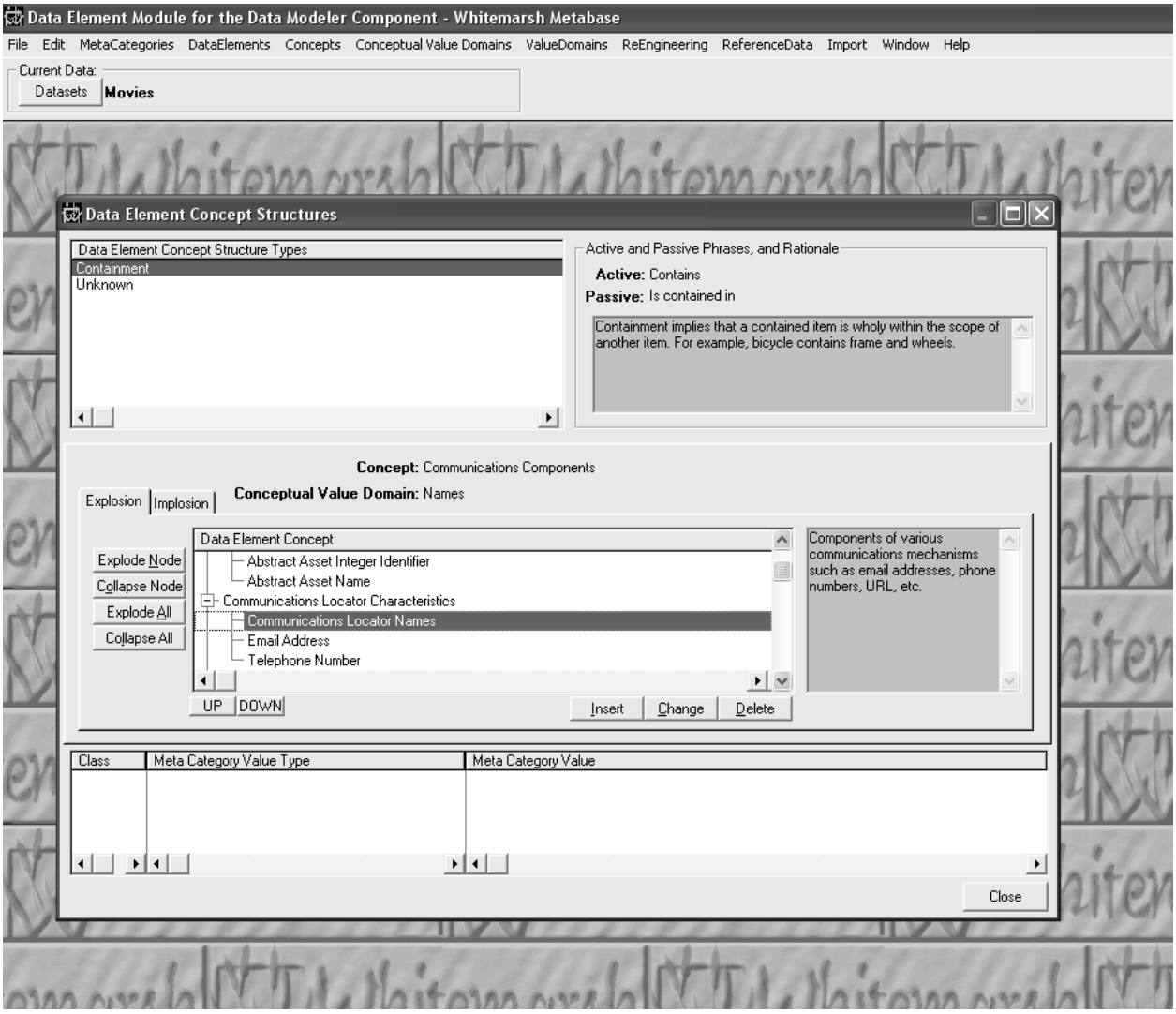

Figure 21. Data Element Concept Structures.

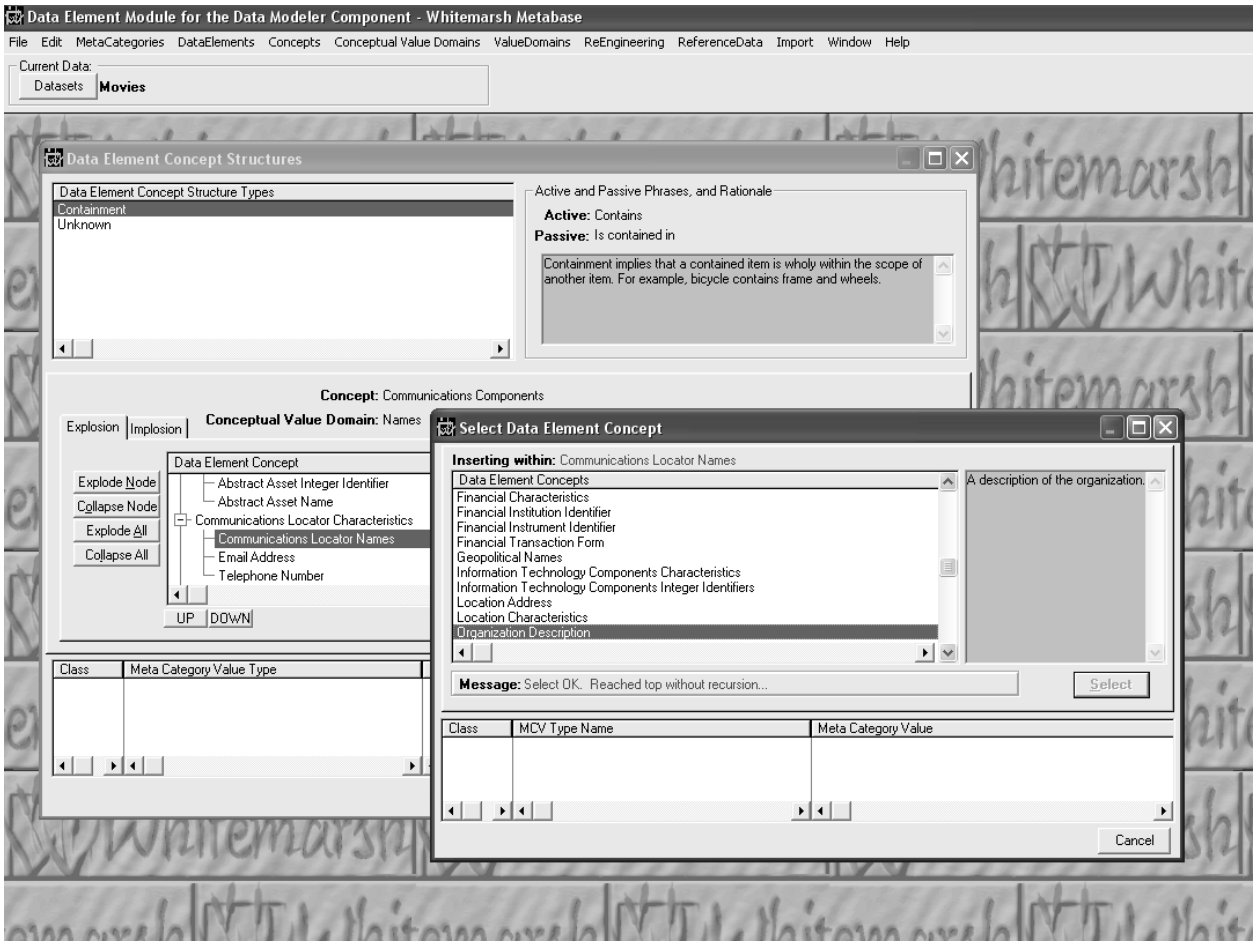

Figure 22. Selecting a Data Element Concept for a Data Element Concept Structure.

6.2.1.3.4 Data Element Concept Structure Types

The data element concept structure type is a way of distinguishing one data element concept structure from another. Figure 23 presents the current list. If two different data element concept hierarchies are interconnected, it may be that the intersection is distinguished from the others by means of a different data element concept structure type.

Figure 24 presents the data element concept structure type update form. Not only is the name and description of the data element concept structure type provided, so too is the active phrase and the passive phrase. The active phrase is employed by the Whitemarsh metabase system when a down-ward structure is presented. For example, <parent> contains <child 1>, <child 2>, ..., <child n>. The passive phrase enables the reverse phrases to be presented. That is, <child 2> is contained in <parent>.

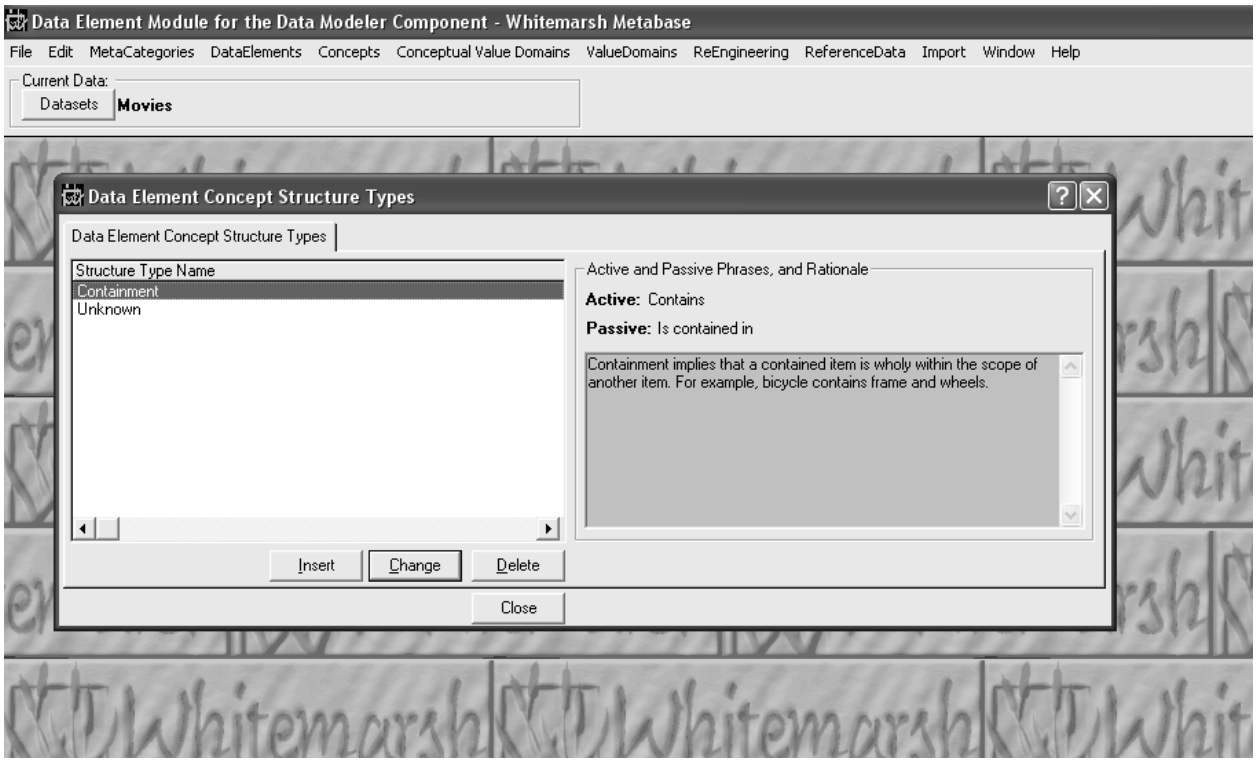

Figure 23. Data Element Concept Structure Types.

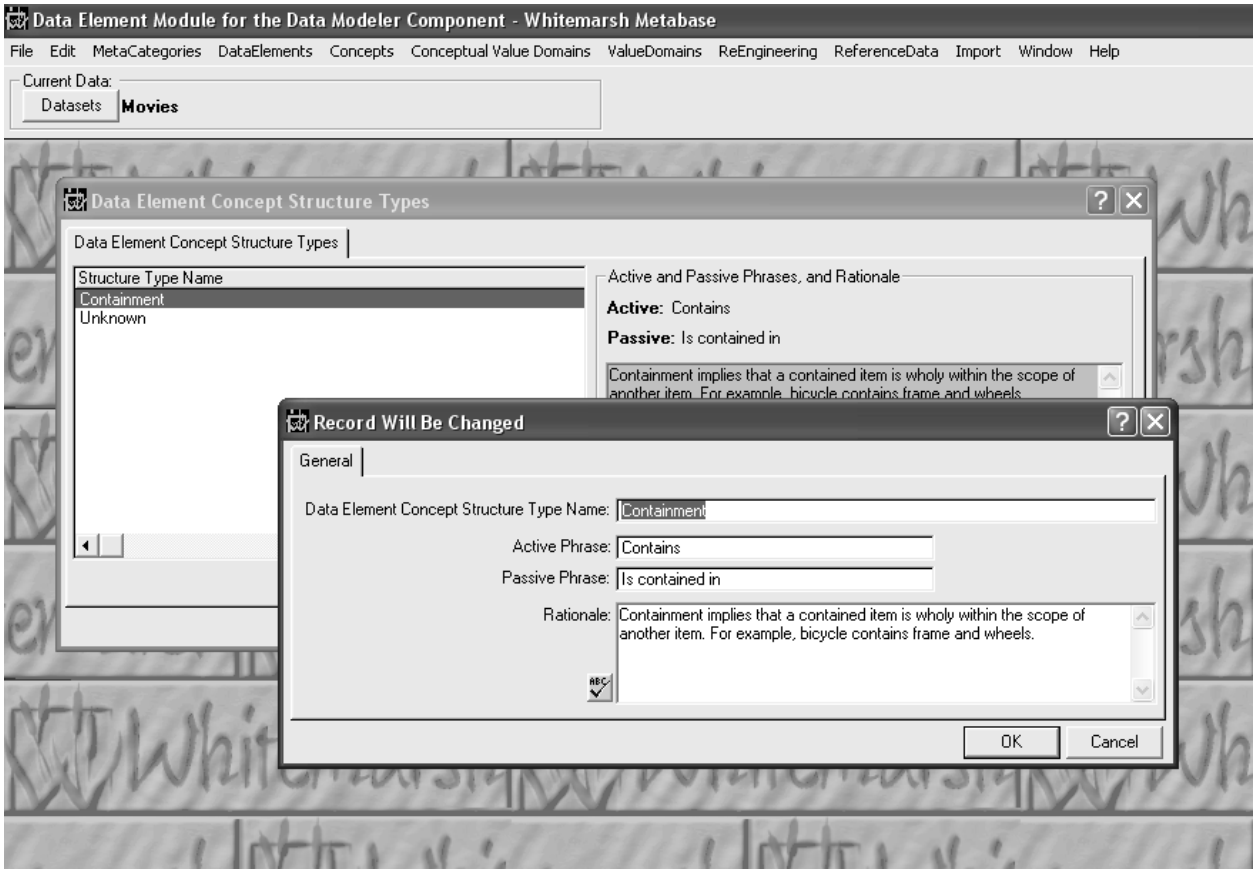

Figure 24. Data Element Concept Structure Type update screen.

6.2.1.3.5 Data Element Concept Meta Category Values

The semantics of data element concept are assigned. Assigning semantics is illustrated in Figure 25. In this assignment process, the appropriate data element concept is highlighted and then tagged. If more than one is tagged, only the first is "remembered." If there are any already assigned semantics then they are shown in the bottom browse window. The right two windows contain the set of meta category value types and type classes, and the meta category values for a particular highlighted meta category value. Only one meta category value from each meta category value type can be tagged. For example, it makes no sense to assign both estimated and projected to the same data element.

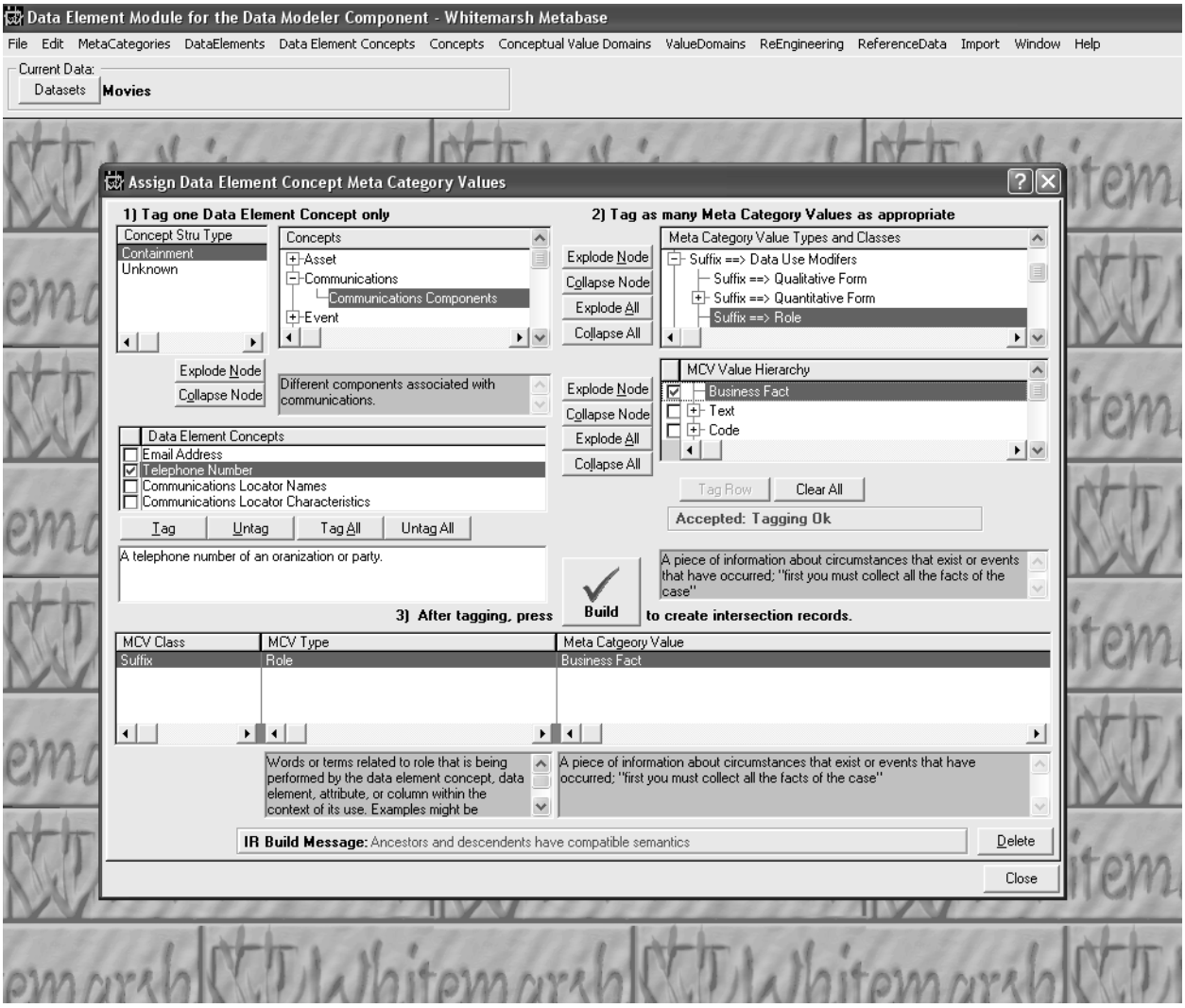

Figure 25. Data Element Concept Meta Category Value Assignment.

Suffix meta category values will then appear at after the data element concept's common business name. Prefix meta category values will appear prior to the data element concept's common business name. The order of appearance is not arbitrary or able to be change by a metabase end-user. The order is specified within the table, meta category type. So, if the geographic meta category value type's sequence number is 2 and the temporal meta category value type's is 3 then regardless of the sequence of tagging, geographic meta category values will always be prefixed into the data element's constructed name before those for the temporal meta category value type.

Once the meta category values are tagged and the build button is pressed, the meta category values are assigned to the data element. The assignment process determines one by one that the tagged meta category values are semantic subsets of those already assigned to a "child" data element, attribute or column. If the tagged semantic is a semantically acceptable then it is assigned. Otherwise the assignment for the offending semantic is rejected.

Once a semantic is assigned, if the data element concept is employed in the creation of a data element then none of the semantics for the data element concept can be deleted. New ones can be added, but none can be deleted.

When ever semantics are added, the data element concept's name remains the same until it is reset in the data element concept update screen.

6.2.1.3.6 Data Element Concepts and Data Elements

Figure 26 presents the data elements associated with data element concepts. This a "report" screen.

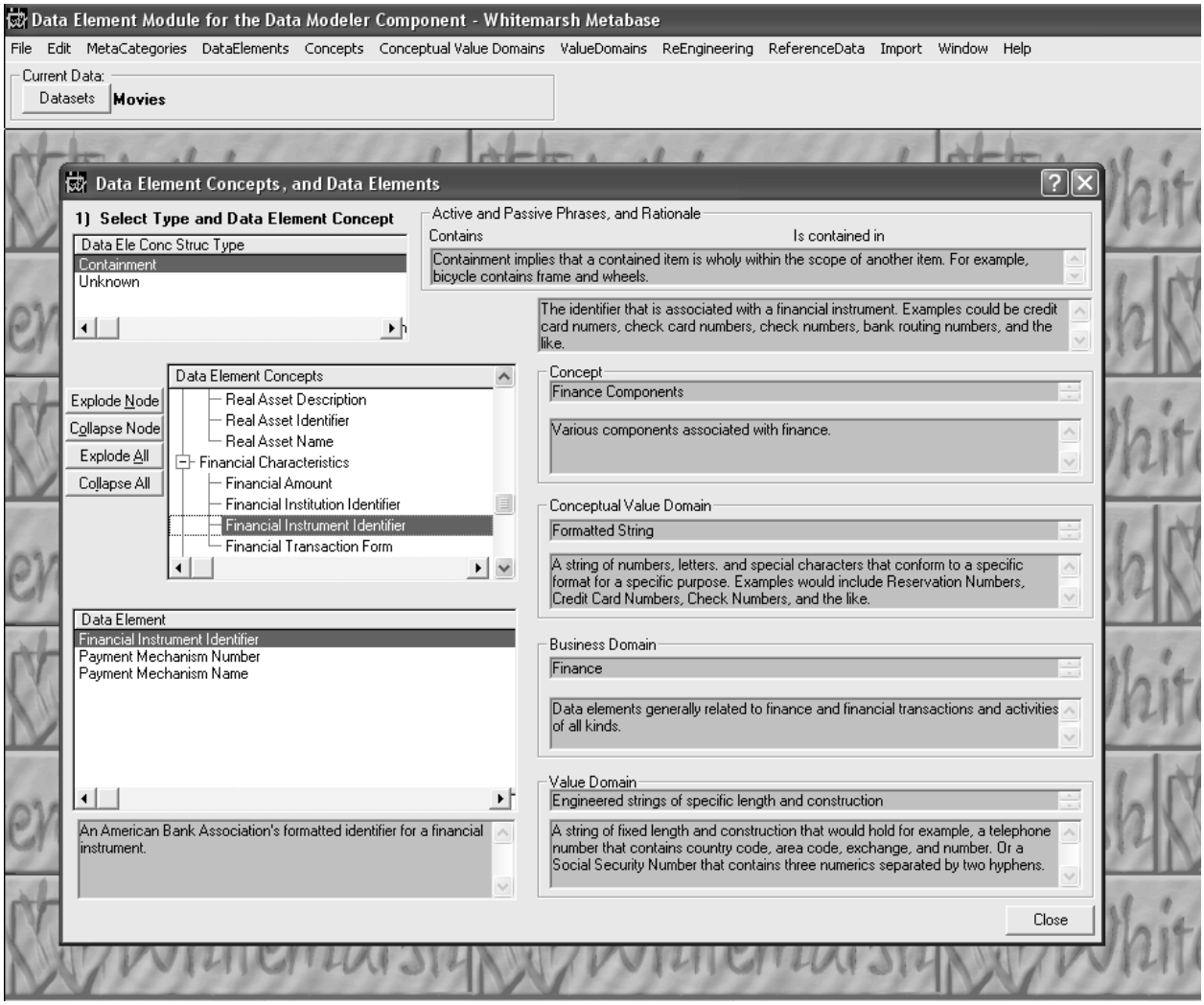

Figure 26. Data Element Concepts and Data Elements.

6.2.1.4 Compound Data Elements

Compound data elements consist of the following:

- Compound data elements
- Compound data element structures
- Compound data element structure types
- Assignment of data elements to a compound data element

6.2.1.4.1 Compound Data Elements

Compound data elements are a named collection of contained data elements. Simple compound names are simple hierarchies like full name, which includes first name, middle name, and last name. More complex compound names are bills-of materials. For example, there might be the following address components:

- Organization name
- Person name
- Building name
- Suite and/or floor identification
- Street Address
- City State Postal Code

Each of these address components might be composed of multiple data elements. For a typical residential address, the components needed are just 2, 5, and 6. But for a simple business address it might be 1, 5, and 6. But a more complex address might involve all six. Because there are different structures to the address types, the data structure is really a bill of materials.

Compound data elements do not map to attributes of entities, or to columns of tables. Only data elements map. Instead, compound data elements map to database application view columns within the metabase's data modeler's view data model. The primary purpose of a compound data element is to understand the data values that exist within database applications.

Figure 27 presents a list of compound data elements. Not all of these compound data elements map to data elements because they are intermediate components within the compound data element's bill of materials.

In Figure 27, the compound data element, Person Address maps to the data elements" Person Name, Geopolitical Name, Street Address, and Postal Code.

Figure 28 presents the update screen for compound data element. The only information required is its name and description.

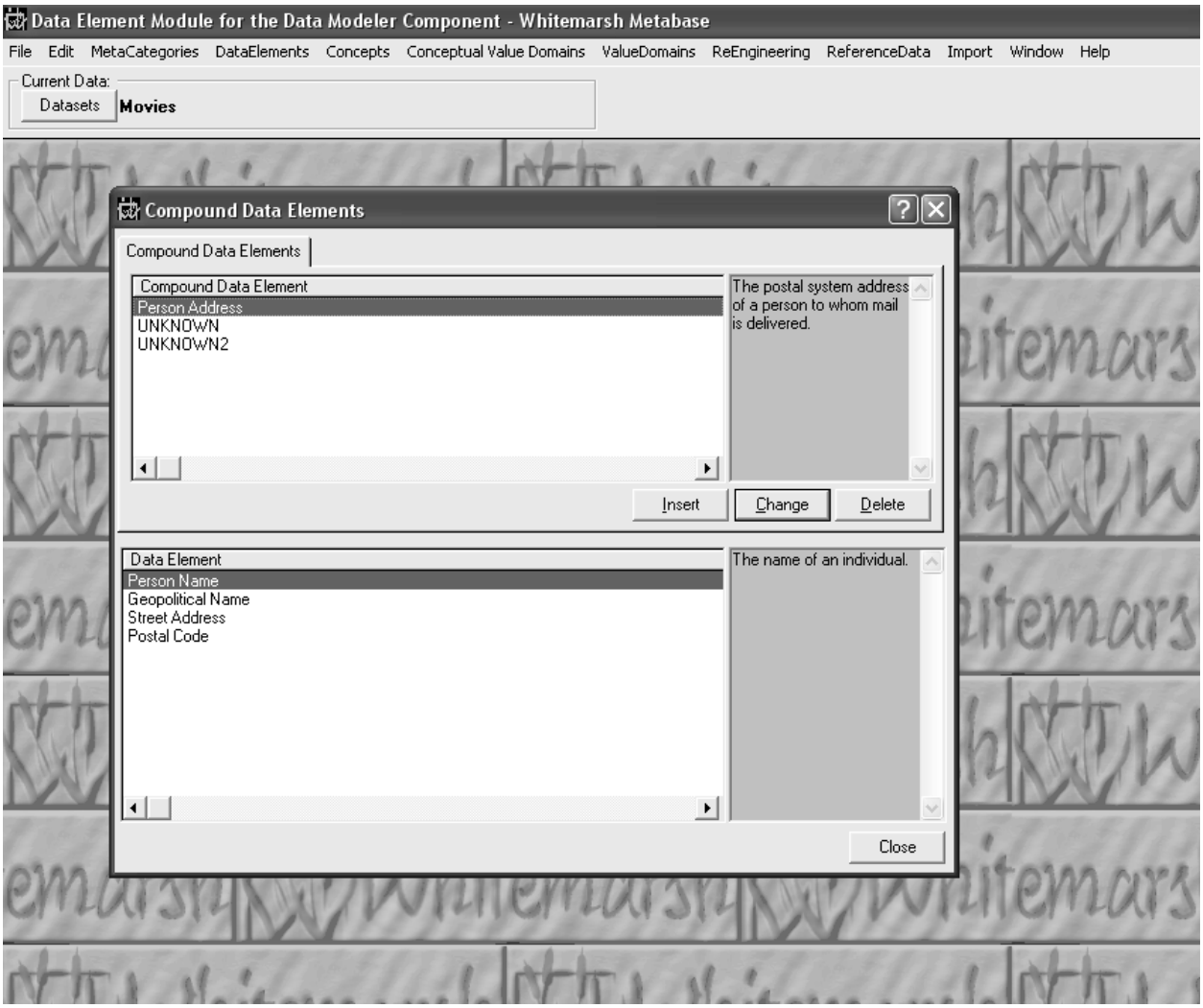

Figure 27. Compound Data Elements.

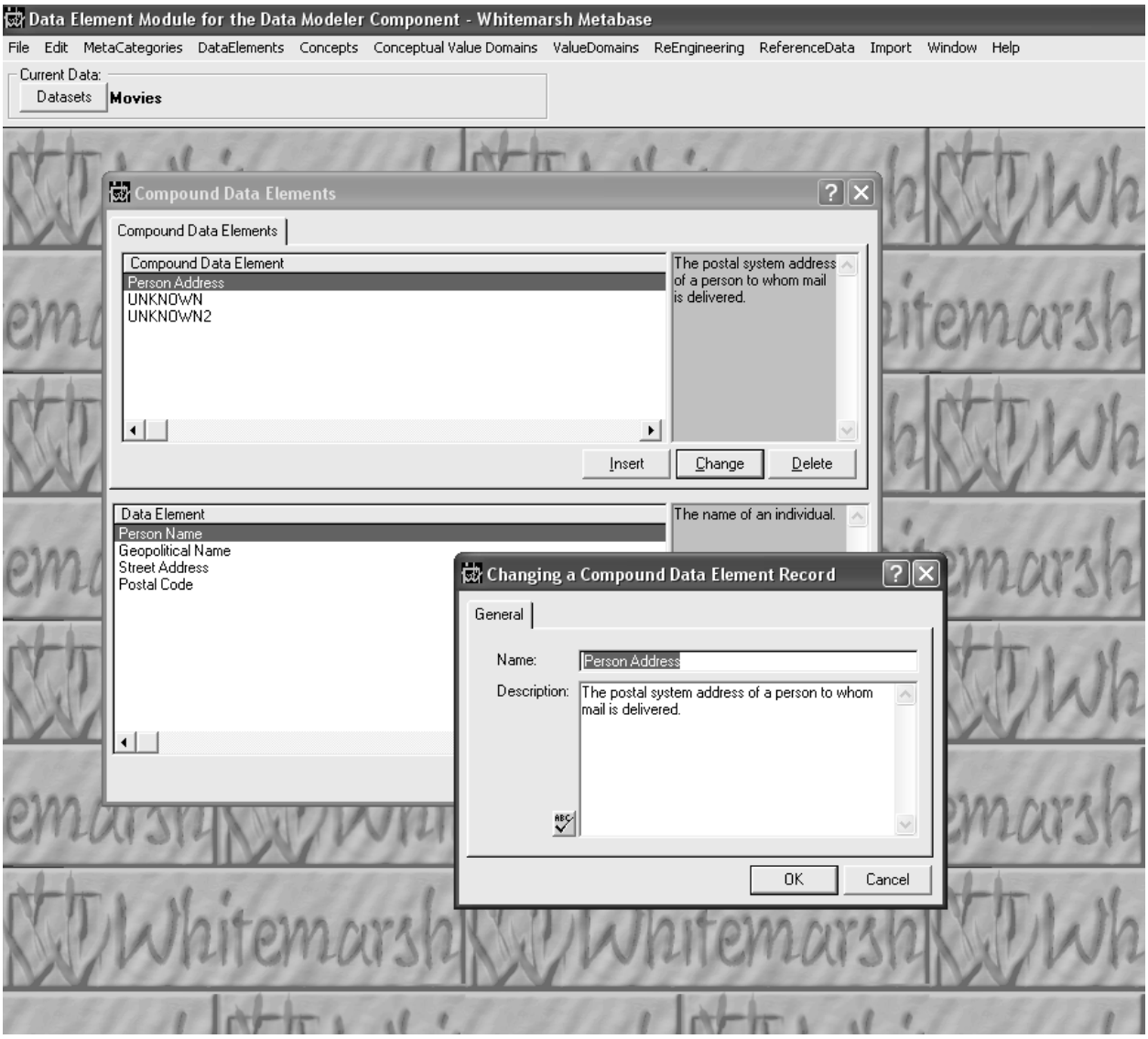

Figure 28. Compound Data Element update screen.

6.2.1.4.2 Compound Data Element Structures

Compound data elements can exist singly or in hierarchies, or networks. In the last case, compound data elements form a traditional bill of materials data structure. Users are encouraged to view the Whitemarsh application WisBOM, Whitemarsh Bill of Materials. This application can be downloaded from the Whitemarsh website and it contains it's own data examples, supporting diagrams and documentation. This application is provided as a training device.

Figure 29 shows a existing set of compound data element structures. In this example, the same compound data element, Person Address, is contained in two other compound data elements, business address and residential address. Hence, the bill of materials.

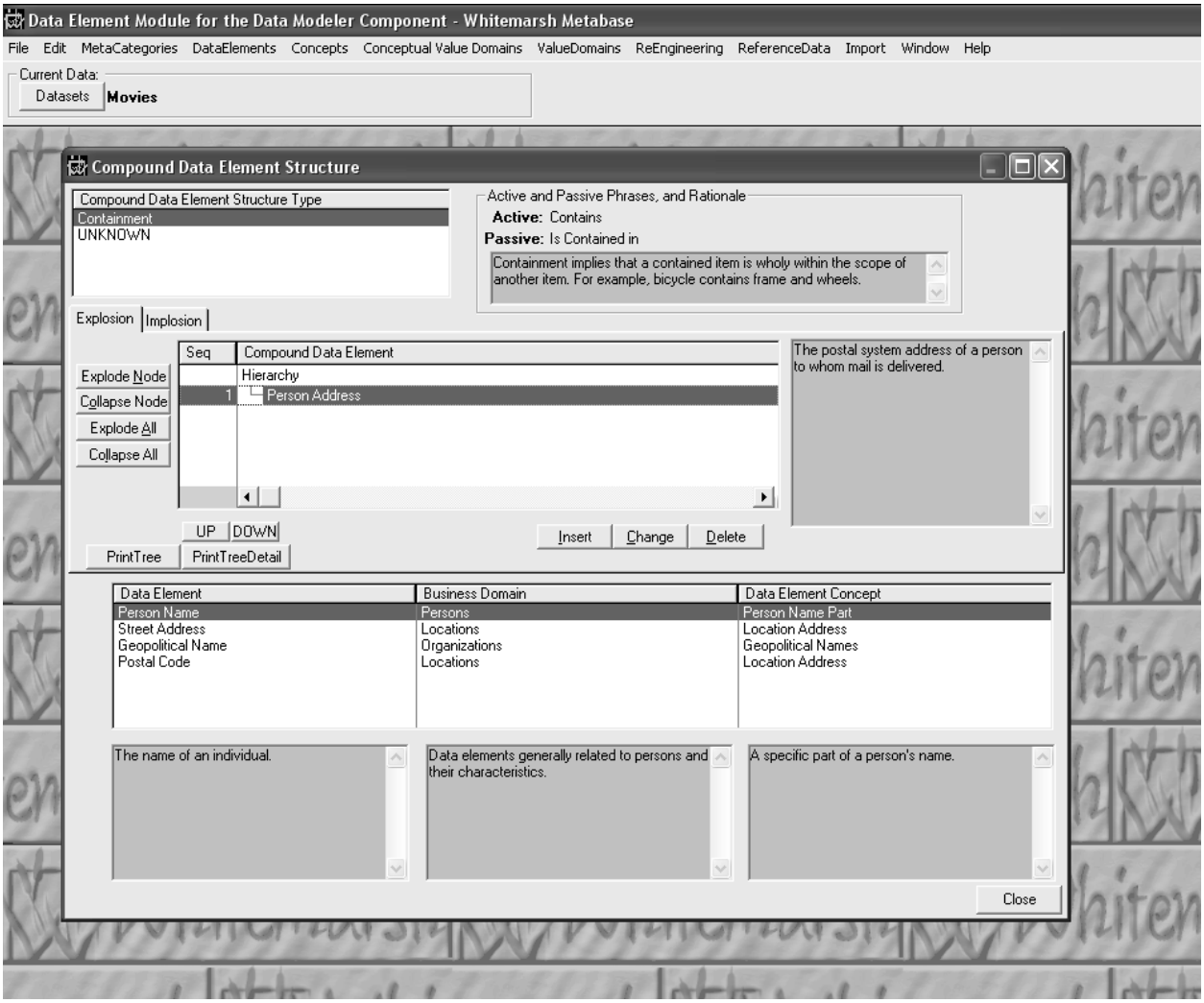

Figure 29. Compound Data Elements within Compound Data Element Structures.

When a compound data element's name is highlighted and the Insert button is pressed, a screen like Figure 30 is presented. The highlighted and selected compound data element, is then included in the existing set. Automatically excluded are recursions, twins, and infinite recursions.

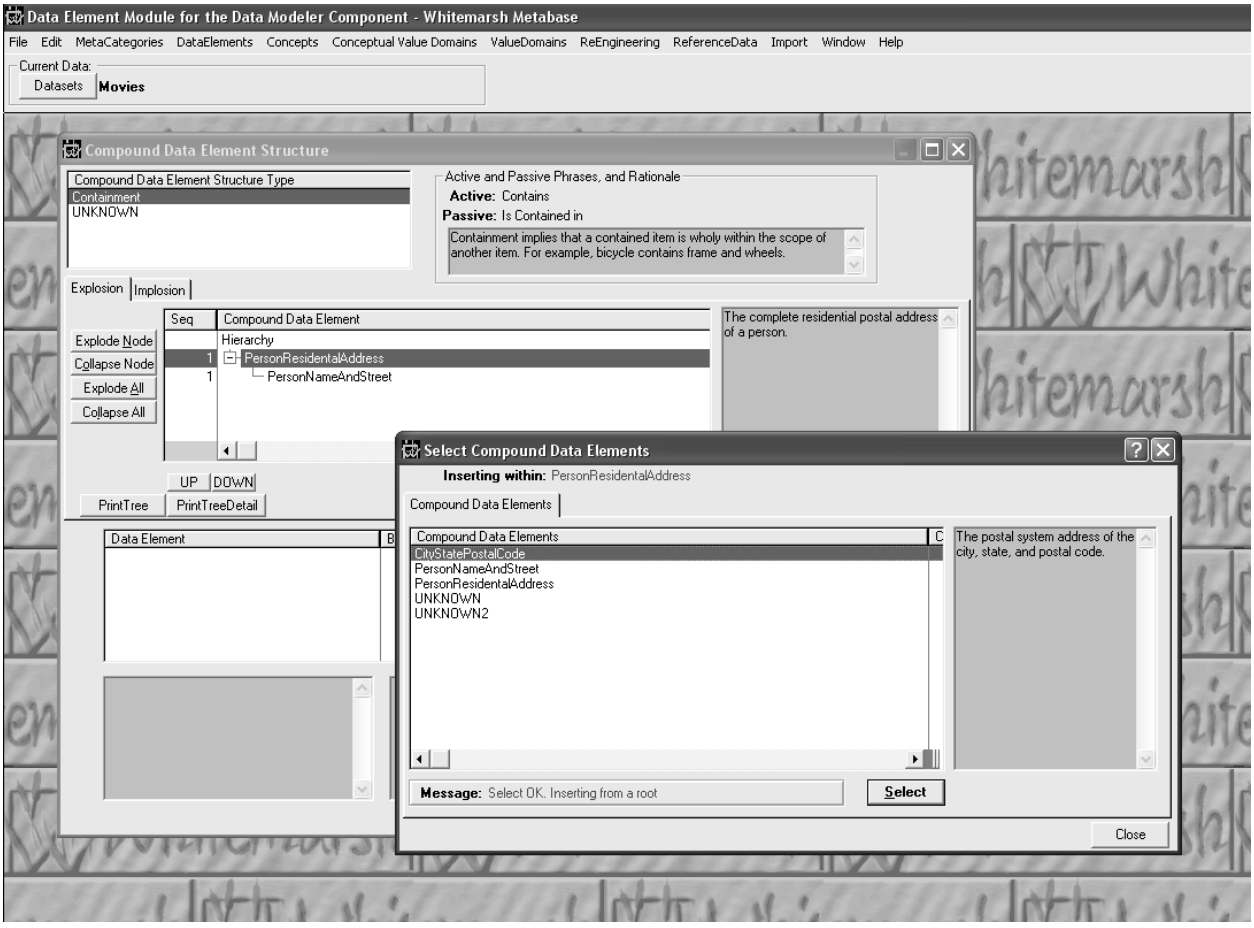

Figure 30. Inserting a Compound Data Element within a Compound Data Element Structure.

6.2.1.4.3 Compound Data Element Structure Types

The compound data element structure type is a way of distinguishing one compound data element concept structure from another. Figure 31 presents the current list. If two different compound data element hierarchies are interconnected, it may be that the intersection is distinguished from the others by means of a different compound data element structure type.

Figure 32 presents the compound data element structure type update form. Not only is the name and description of the compound data element structure type provided, so too is the active phrase and the passive phrase. The active phrase is employed by the Whitemarsh metabase system when a down-ward structure is presented. For example, <parent> contains <child 1>, <child 2>, ..., <child n>. The passive phrase enables the reverse phrases to be presented. That is, <child 2> is contained in <parent>.

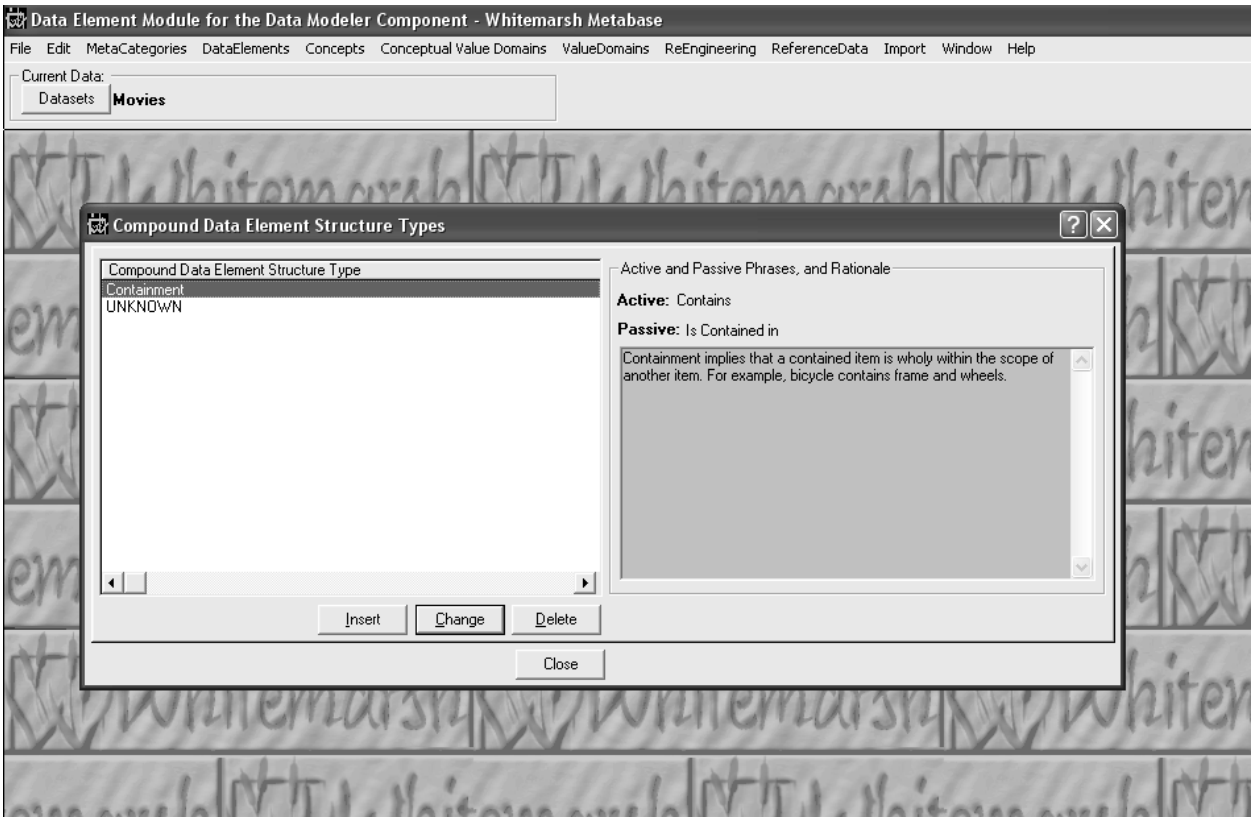

Figure 31. Compound Data Element Structure Types.

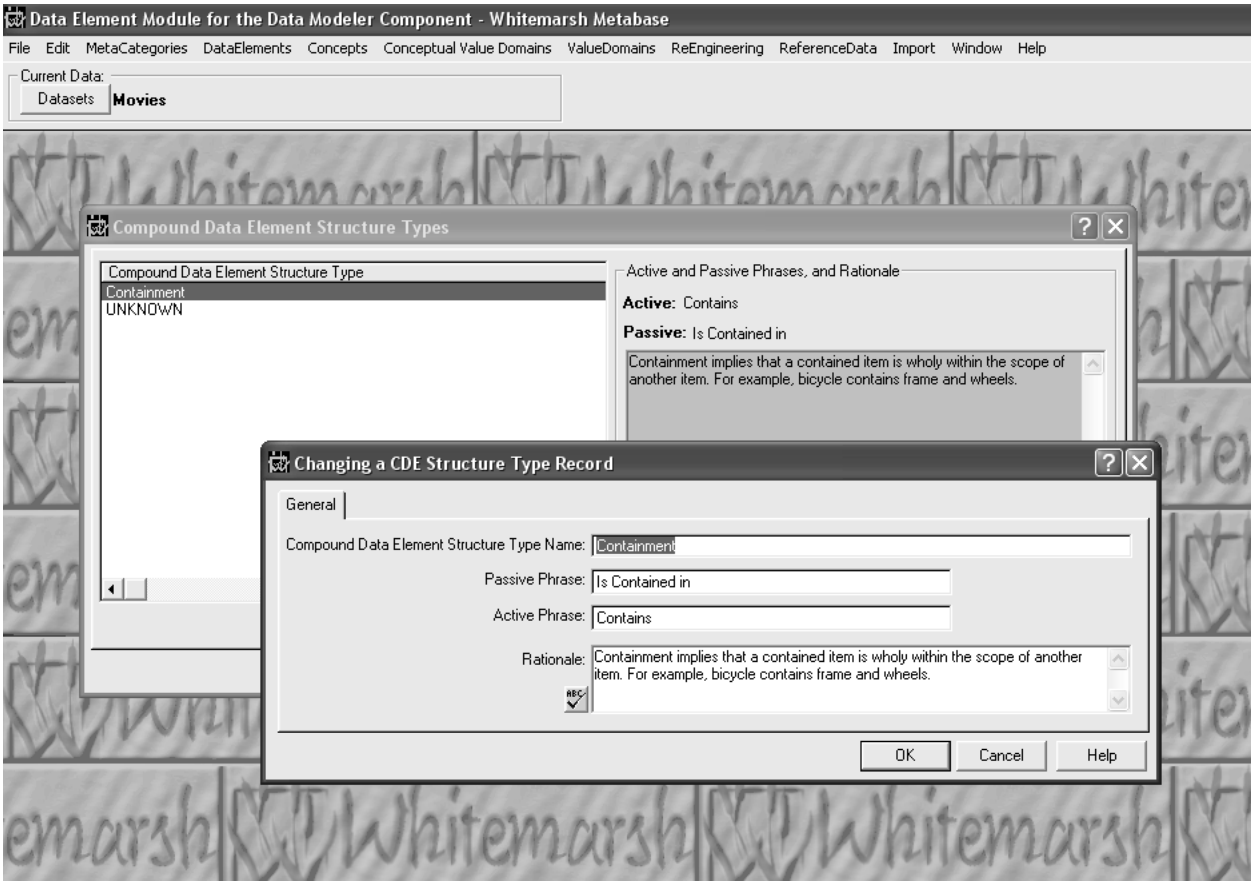

Figure 32. Compound Data Element Structure Type update screen.

6.2.1.4.4 Assignment of Data Elements to a Compound Data Element

The process of assigning data elements to a compound data element is straight forward. Figure 33 presents the window. A compound data element is tagged in the upper left window. Then one or more data elements are tagged in the upper right window. Then the build buttons is pressed. Once the records are build, their relative sequence may be changed via the up and down buttons.

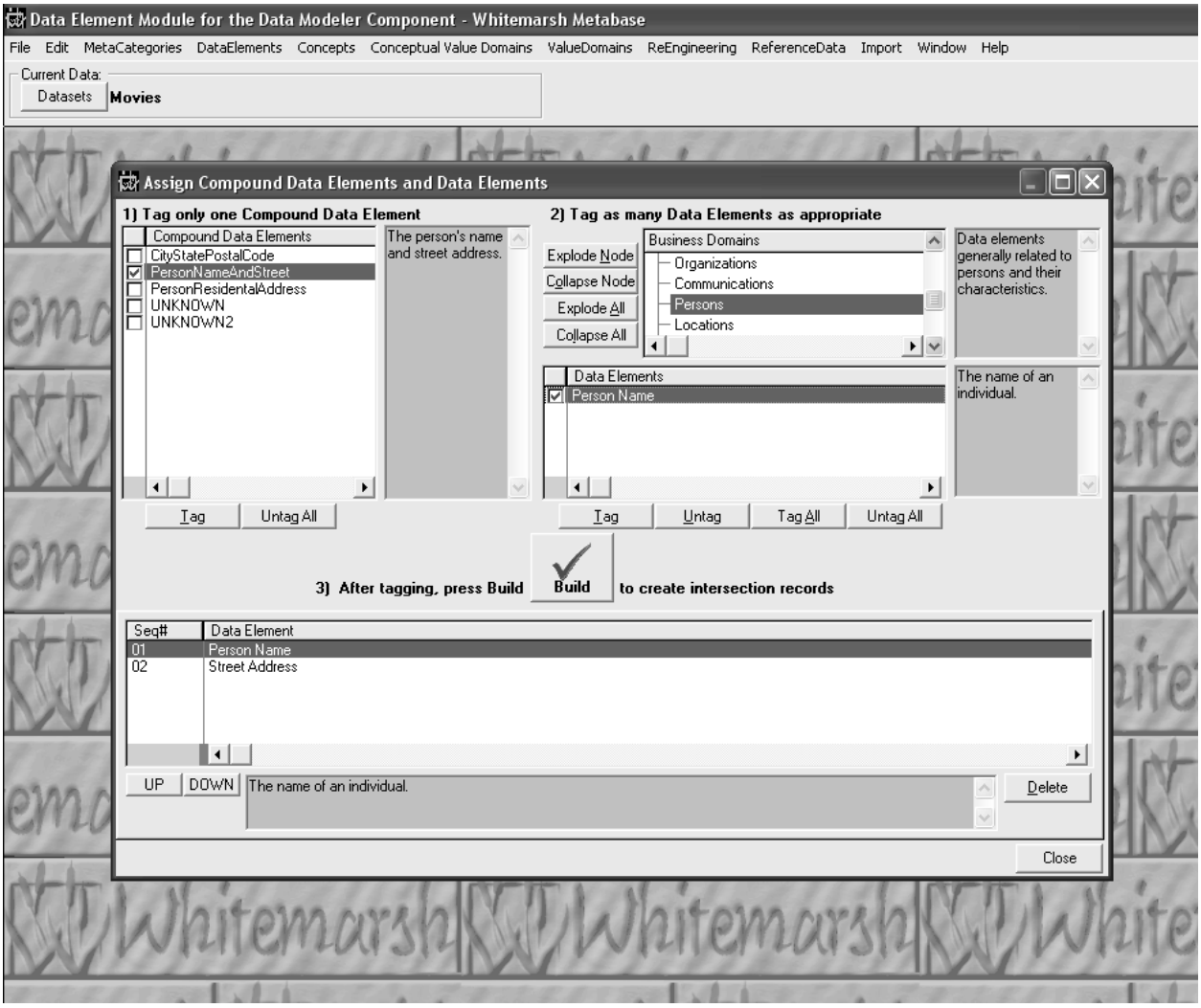

Figure 33. Assigning Data Elements to a Compound Data Element.

6.2.1.5 Derived Data Elements

Derived data elements consist of the following:

- Derived data elements
- Assign derived data elements to data elements
- Assign derived data elements to compound data elements

6.2.1.5.1 Derived Data Elements

A derived data element results from the execution of a defined data transformation process on a collection of data elements. Derived data elements do not map to attributes of entities, or to columns of tables. Only data elements map. Instead, derived data elements map to database application view columns within the metabase's data modeler's view data model. The primary purpose of a derived data element is to understand the data values that exist within database applications. Additionally, a compound data element may map to one or derived data elements. For example, an enterprise may have a compound data element, united inches, which it in turn is the sum of a container's length plus width, plus height, then derived dat element, average united inches.

Figure 34 presents a list of derived data elements. This particular derived data element, age is derived from an operation against two data elements, person birth date and non-specific date. To add, delete, or change a derived data element press the appropriate button. Figure 35 presents the update screen for updating a derived data element. The description should contain the necessary information for implementors to produce the desired result.

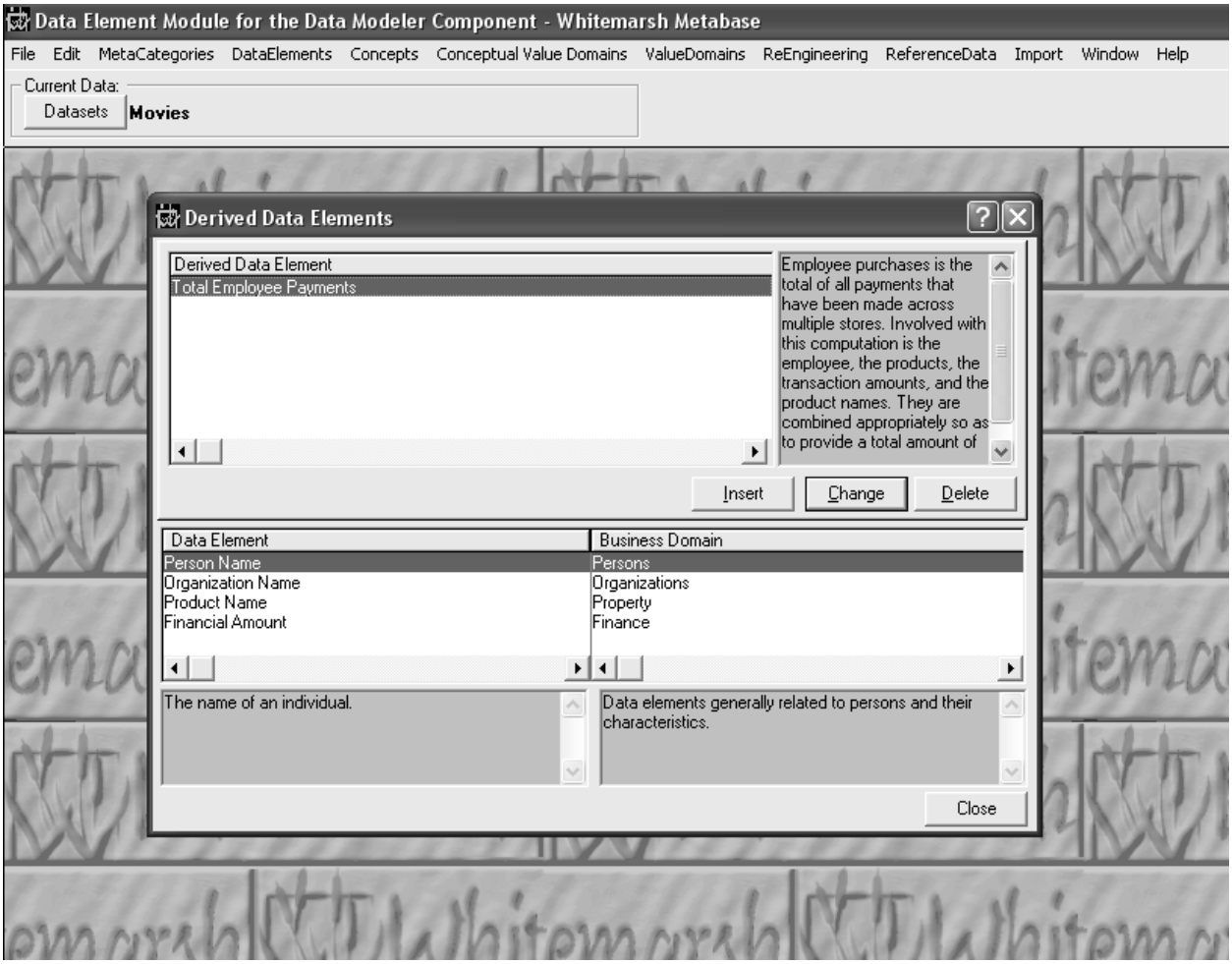

Figure 34. Derived Data Elements and their contained data elements.

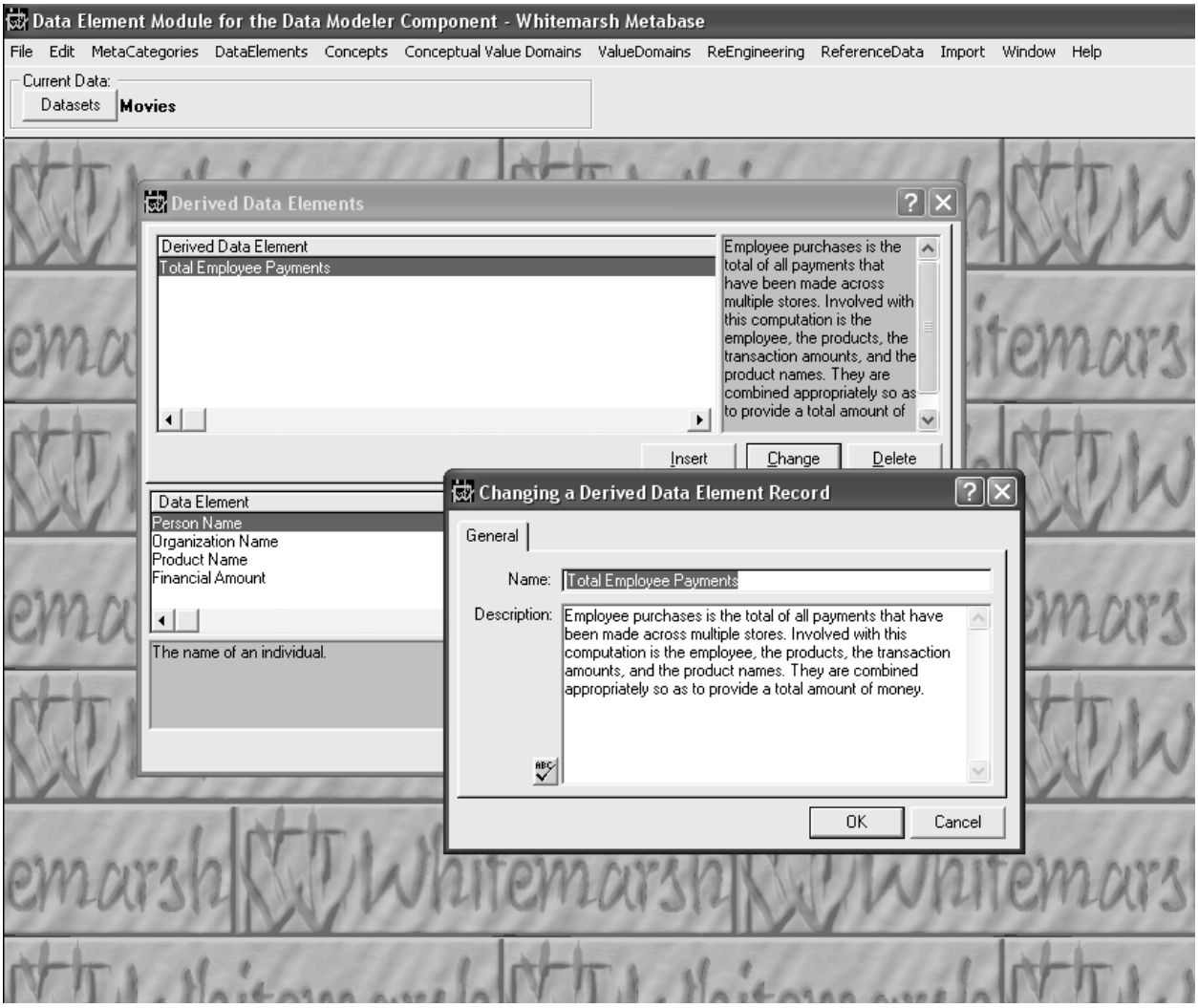

Figure 35. Derived Data Element update screen.

6.2.1.5.2 Assign Derived Data Elements to Data Elements

The process of assigning data elements to a derived data element is straight forward. Figure 36 presents the window. A derived data element is tagged in the upper left window. Then one or more data elements are tagged in the upper right window. Then the build buttons is pressed. Once the records are build, their relative sequence may be changed via the up and down buttons.

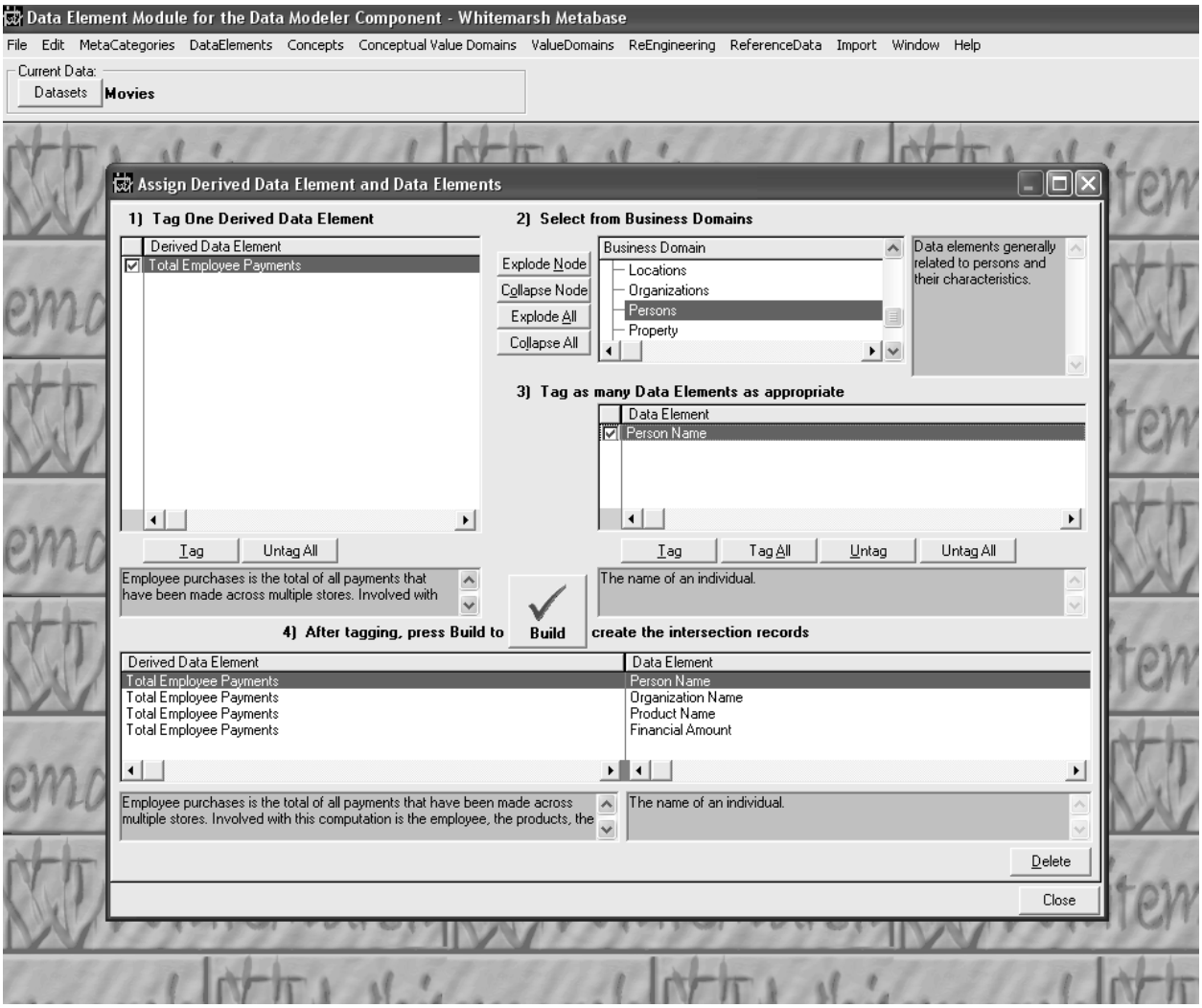

Figure 36. Assigning Data Elements to a Derived Data Element.

6.2.1.5.3 Assign Derived Data Elements to Compound Data Elements

The process of assigning compound data elements to a derived data element is straight forward. Figure 37 presents the window. A derived data element is tagged in the upper left window. Then one or more compound data elements are tagged in the upper right window. Then the build buttons is pressed. Once the records are build, their relative sequence may be changed via the up and down buttons.

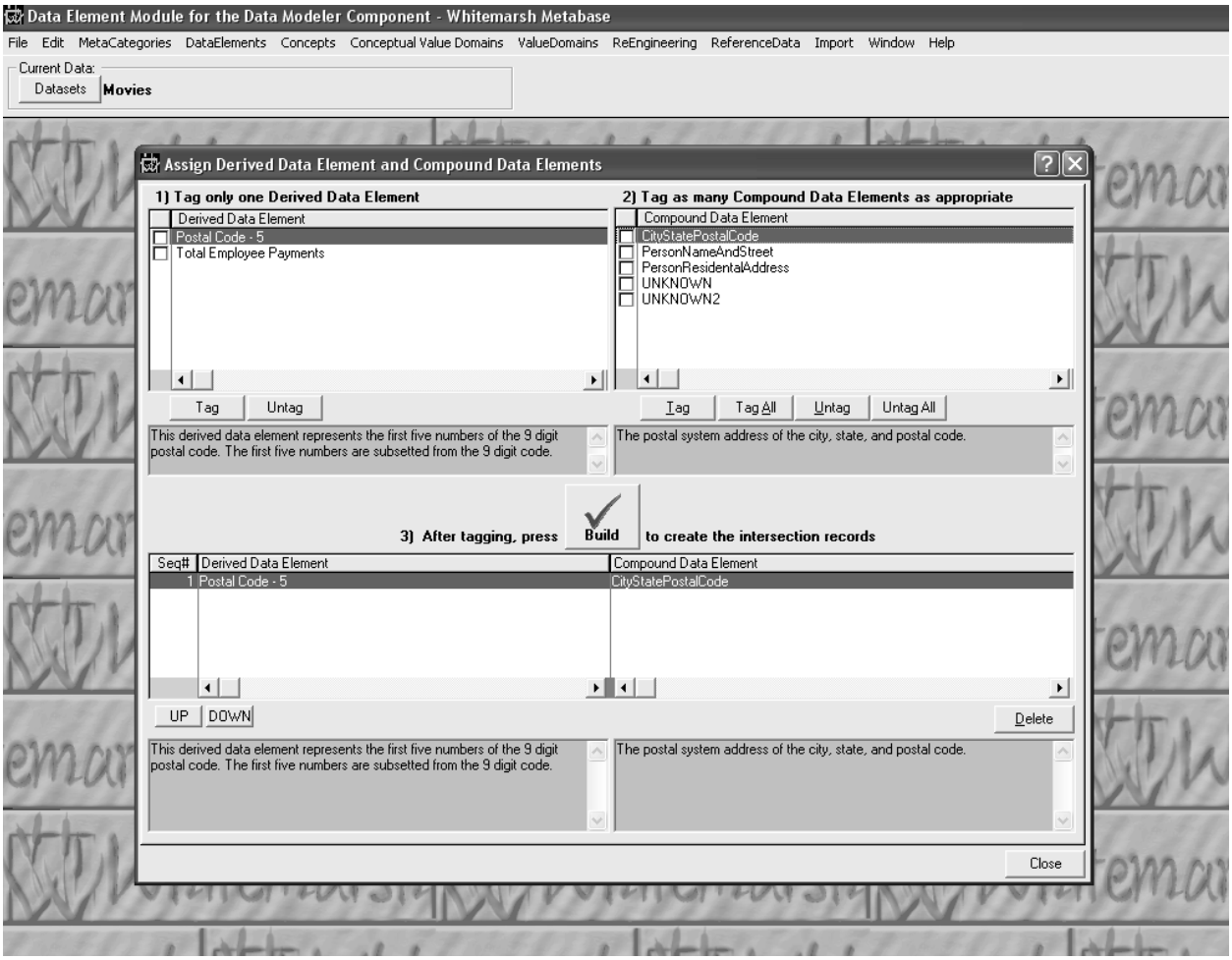

Figure 37. Assigning Derived Data Elements to Compound Data Elements.

6.2.1.6 Reverse Engineering

While the collected set of meta data stored in the metabase is best accomplished in a top-down fashion, that is neither practical nor realistic. Most organizations already have existing databases that need to be reverse engineered from operational data models to implemented data models to specified data models and their columns and attributes reverse engineered into common data elements.

In addition to reverse engineering, there is also a need for example to re-assign columns or attributes to different data elements, or data elements to different business domains. Included in the reverse engineering are thus:

- Reassign data elements to business domains
- Reassign data elements to data element concepts
- Reassign data elements to value domains
- Reassign data element concepts to concepts
- Reassign data element concepts to conceptual value domains
- Reassign value domains to conceptual value domains
- Promote data elements to data element concepts
- Promote data element concept to concept

6.2.1.6.1 Reassign Data Elements to Business Domains

Figure 38 presents the screen for reassigning one or more data elements from one business domain to another. The reassignment consists of tagging one or more data elements in the left window, and then tagging one business domain in the right window. Then press the re-assign button at the bottom. The reassigned data elements will then be re-displayed with their newly assigned business domains.

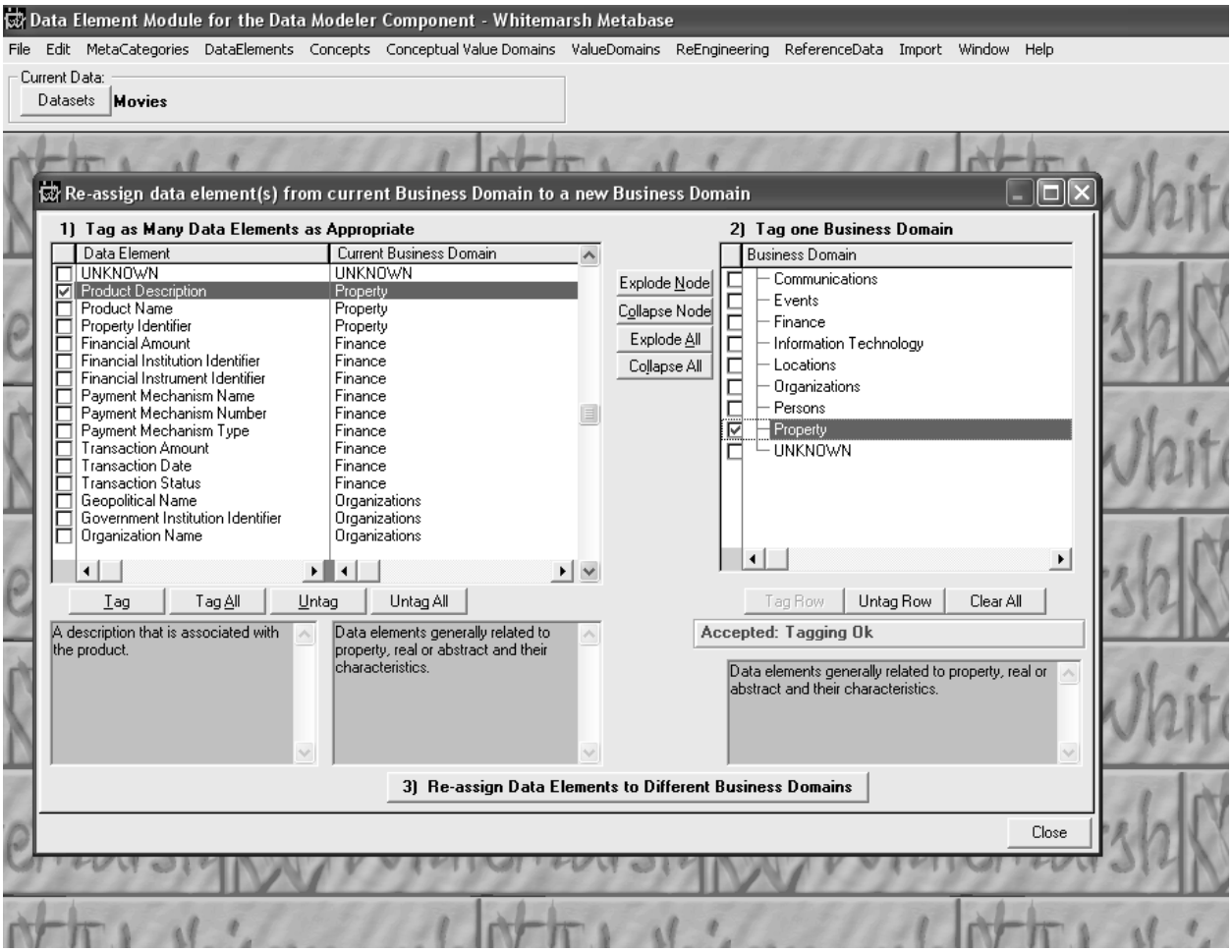

Figure 38. Reassigning Data Elements to a different Business Domain.

6.2.1.6.2 Reassign Data Elements to Data Element Concepts

Figure 39 presents the screen for reassigning one or more data elements from one data element concept to another. The reassignment consists of tagging one or more data elements in the left window, and then tagging one data element concept in the right window. Then press the re-assign button at the bottom. The reassigned data elements will then be re-displayed within their newly assigned data element concepts.

A data element's assignment to a data element concept will be successful if and only if an already assigned value domain references the same conceptual value domain as the to be assigned data element concept. The quickest way to avoid this problem is to first reassign the data element's value domain to unknown, then re-assign the data element concept, then reassign a value domain that is associated with the newly assigned data element concept's conceptual value domain.

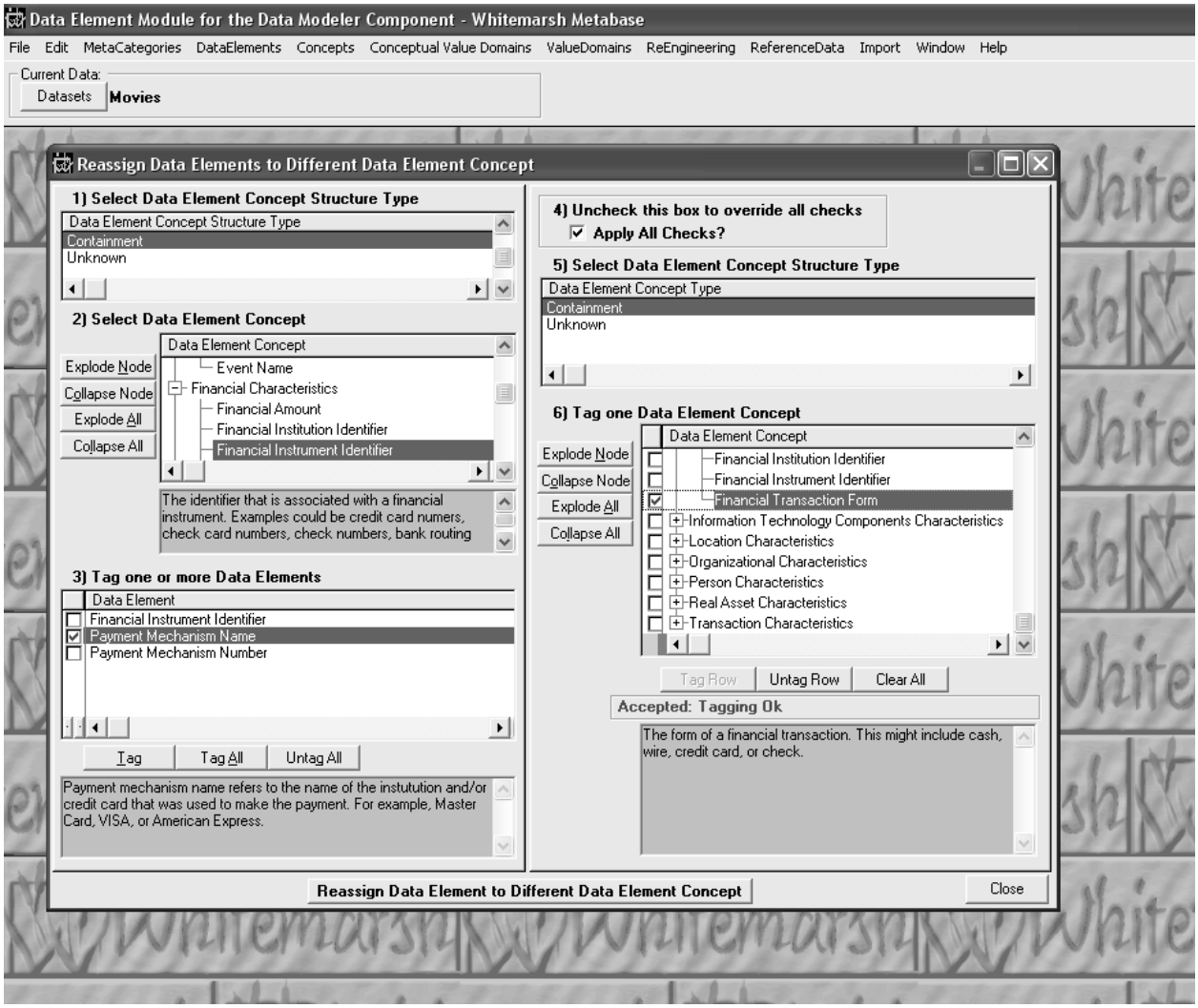

Figure 39. Reassigning a Data Element to a different Data Element Concept.

6.2.1.6.3 Reassign Data Elements to Value Domains

Figure 40 presents the screen for reassigning one or more data elements from one value domain to another. The reassignment consists of tagging one or more data elements in the left window, and then tagging one value domains in the right window. Then press the re-assign button at the bottom. The reassigned data elements will then be re-displayed with their newly assigned value domains.

A data element's assignment to a value domain will be successful if and only if the already assigned data element's data element concept is assigned the same conceptual value domain as the to be assigned value domain's conceptual value domain.

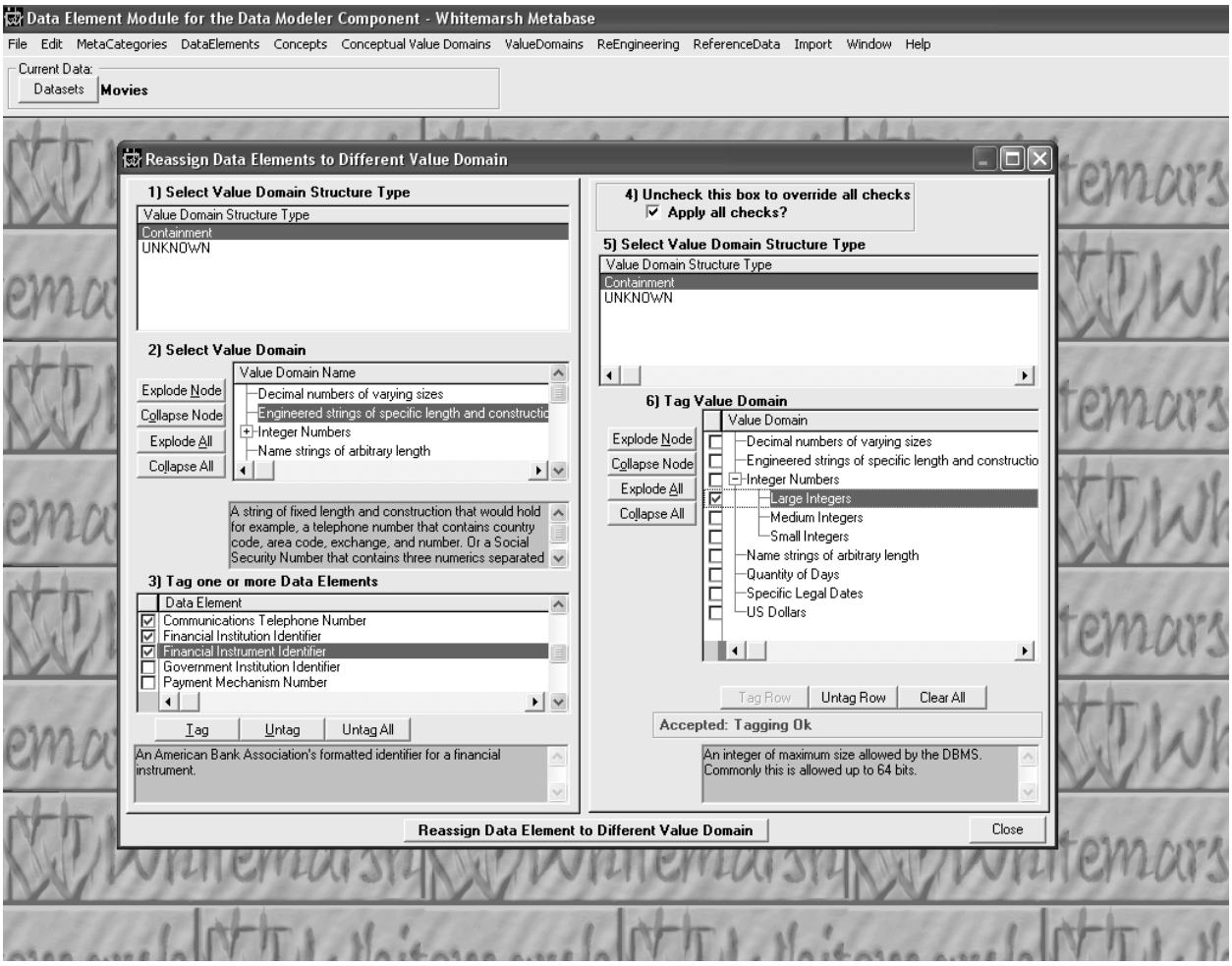

Figure 40. Reassigning data elements to a different value domain.

6.2.1.6.4 Reassign Data Element Concepts to Concepts

Figure 41 presents the screen for reassigning one or more data element concepts from one concept to another. The reassignment consists of tagging one or more data element concepts in the left window, and then tagging one concept in the right window. Then press the re-assign button at the bottom. The reassigned data element concepts will then be re-displayed with their newly assigned concept.

A data element's assignment to a different concept will be successful if and only if the data element concept is not already used for a data element.

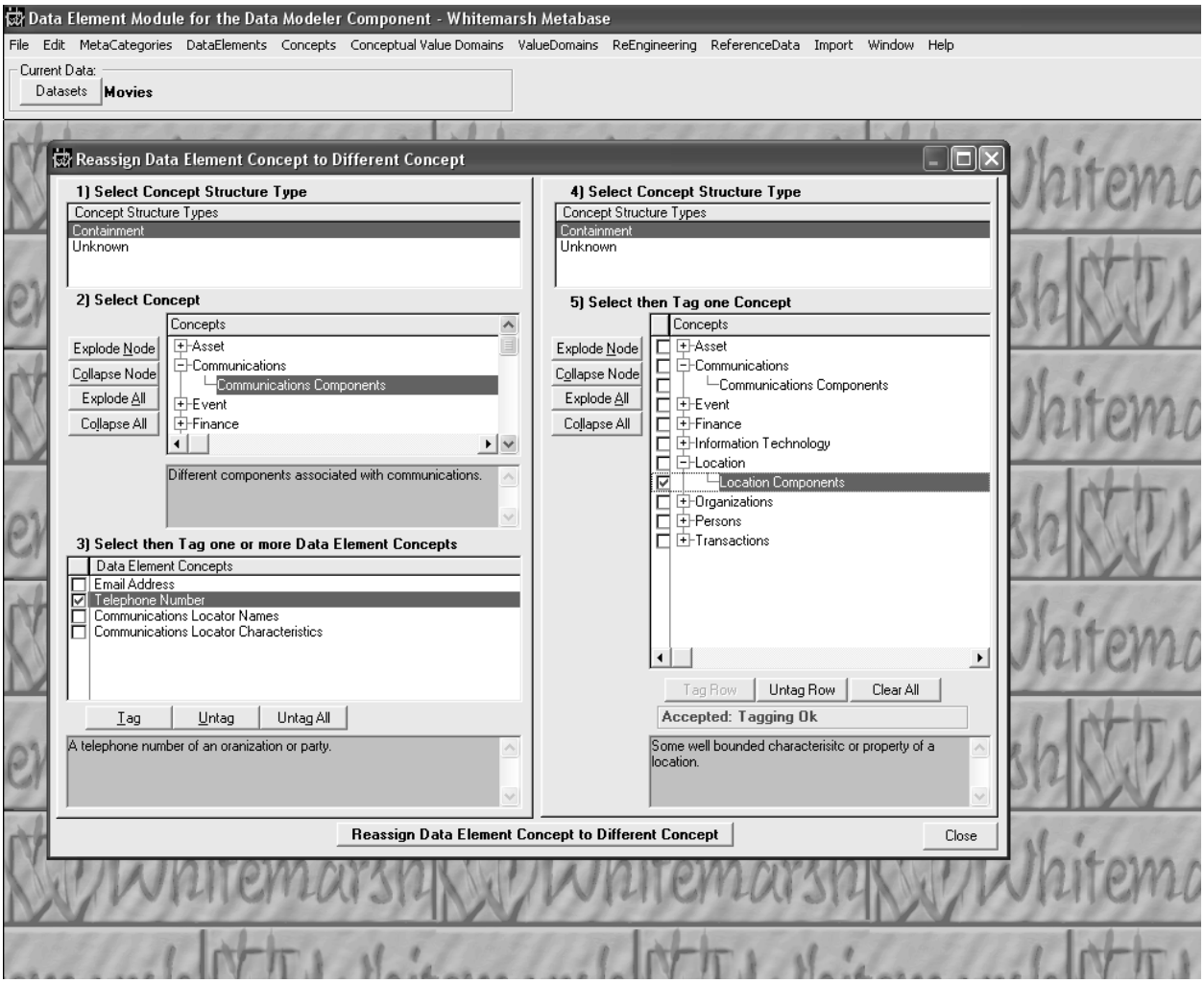

Figure 41. Reassign Data Element Concept to Concept.

6.2.1.6.5 Reassign Data Element Concepts to Conceptual Value Domains

Figure 42 presents the screen for reassigning one or more data element concepts from one conceptual value domain to another. The reassignment consists of tagging one or more data element concepts in the left window, and then tagging one conceptual value domain in the right window. Then press the re-assign button at the bottom. The reassigned data element concepts will then be re-displayed with their newly assigned conceptual value domain.

In the event that a data element concept is already assigned to a data element, then the reassignment of a data element concept to a different conceptual value domain is refused. In this case the only re-assignment that is allowed is to the "Unknown" conceptual value domain. This ensures that the single inheritance rule is enforced. That is, that when a data element has a Value

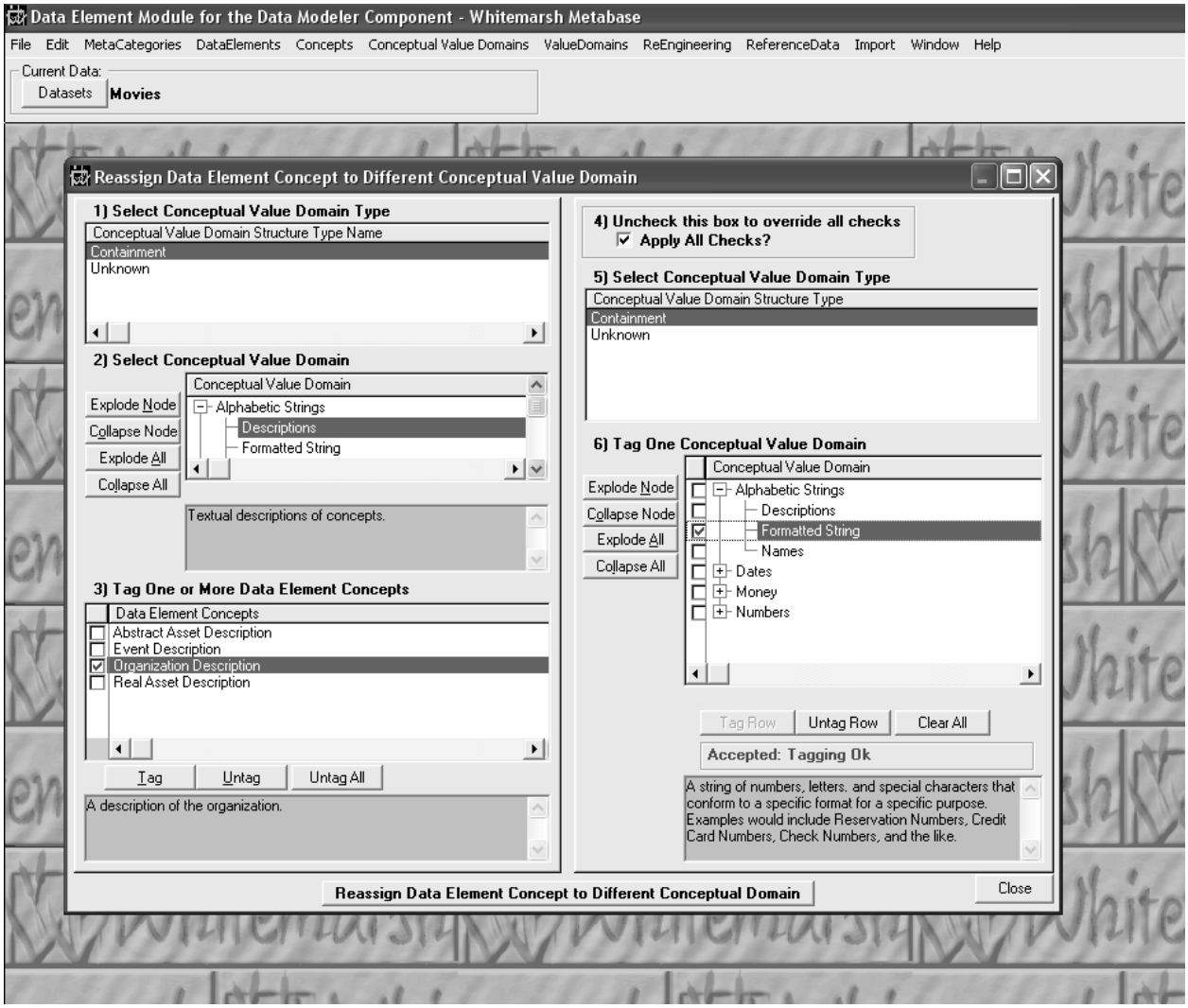

Figure 42. Reassigning a Data Element Concept to a different Conceptual Value Domain.

domain and also a Data Element Concept (which it must have, even if it is "unknown.") then

both the data element's value domain and its data element concept must be from the same conceptual value domain.

6.2.1.6.6 Reassign Value Domain to Conceptual Value Domains

Figure 43 presents the screen for reassigning one or more value domains from one conceptual value domain to another. The reassignment consists of tagging one or more value domains in the left window, and then tagging one conceptual value domain in the right window. Then press the re-assign button at the bottom. The reassigned value domains will then be re-displayed with their newly assigned conceptual value domain.

In the event that a value domain is already assigned to a data element, attribute, column, or DBMS column, then the reassignment of a value domain to a different conceptual value domain is refused. is refused. In this case the only re-assignment that is allowed is to the “Unknown” conceptual value domain. This ensures that the single inheritance rule is enforced. That is, that when a data element has a Value domain and also a Data Element Concept (which it must have, even if it is "unknown.") then both the data element's value domain and its data element concept must be from the same conceptual value domain.

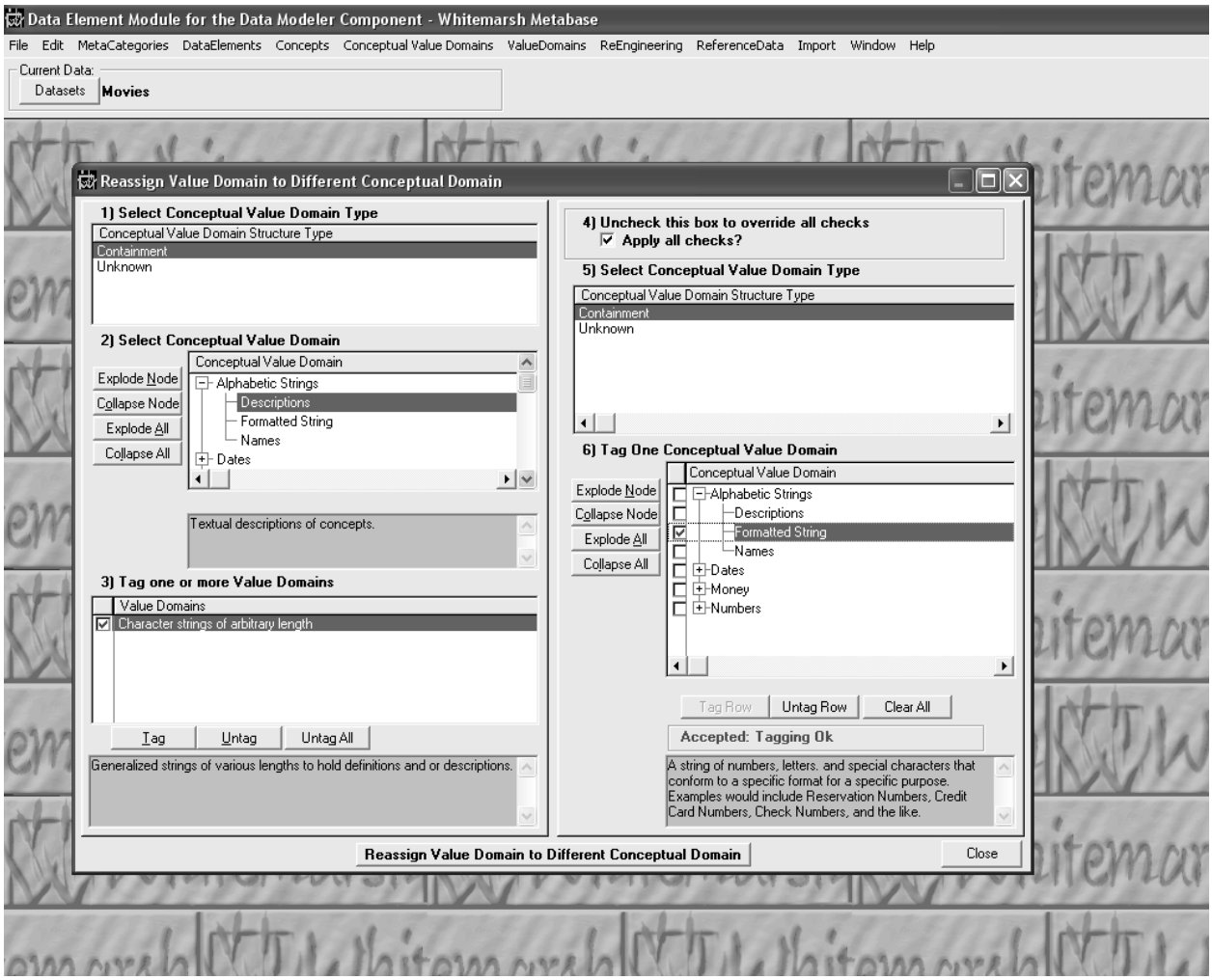

Figure 43. Reassign Value Domain to Conceptual Value Domain.

6.2.1.6.7 Promote Data Elements to Data Element Concept

Figure 44 presents the screen for promoting a data element to be a data element concept. The promotion process consists of highlighting an existing data element and pressing the promotion button. The only data element that are able to be promoted are those that are not already assigned. Not assigned means that they are assigned to "Unknown." At that point, a new data element concept is created from the data element and assigns to that newly created data element concept all the semantics assigned to the promoted data element. The newly created data element concept is assigned the data element concept type, Unknown. Additionally, the data element concept is assigned the Unknown Concept and Conceptual Value Domain. The semantics assigned to the data element are then deleted as they are then inherited from the newly created data element concept. The existing data element is not automatically removed as it may have been the "parent" of one or more attributes or columns. If a delete operation is attempted, and if the data element is not a parent of an attribute or column then the delete process will succeed.

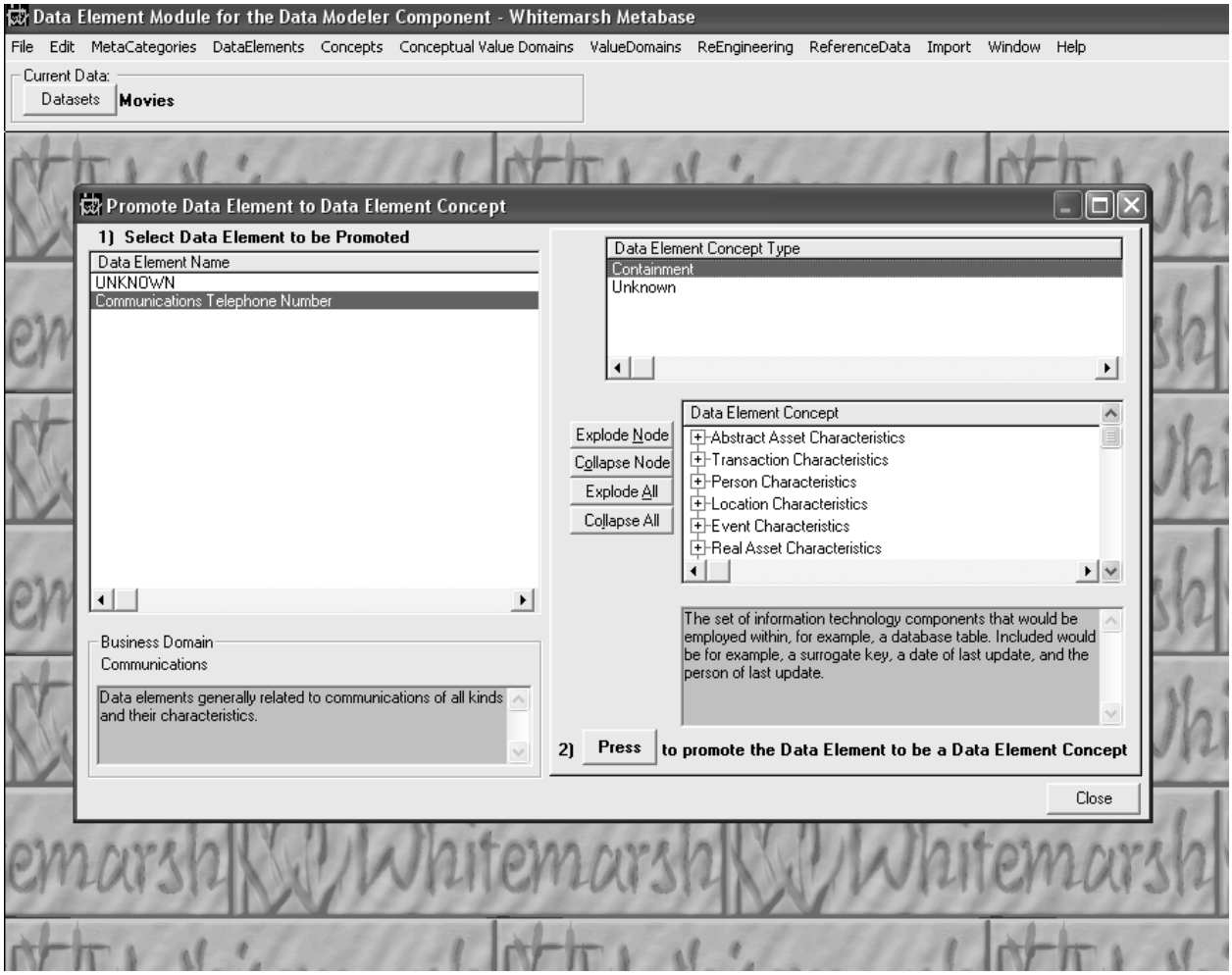

Figure 44. Promote Data Element to Data Element Concept.

6.2.1.6.8 Promote Data Element Concept to Concept

Figure 45 presents the screen for promoting a data element concept to be a concept. The promotion process consists of highlighting an existing data element concept and then pressing the promotion button. The only data element concepts that are able to be promoted are those that are not already assigned. Not assigned means that they are assigned to "Unknown." A new concept is created from the data element, and the new concept then contains as its first member, the data element concept from which it was promoted.

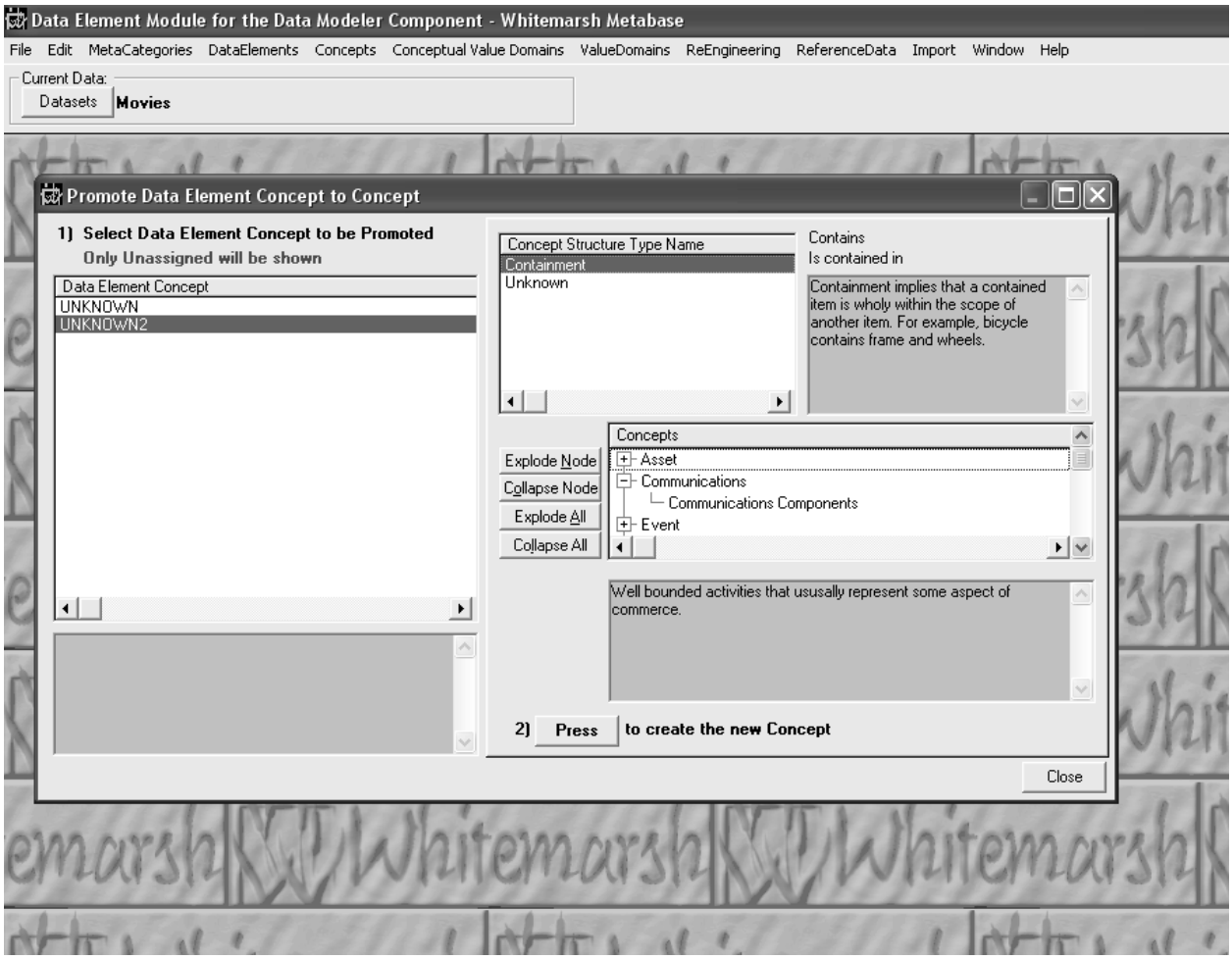

Figure 45. Promotion of a Data Element Concept to a Concept.

6.2.2 Meta Category Value Hierarchies

Meta category value hierarchies are a key source of the overall set of semantics for data element concepts, data elements, attributes and columns. The metabase has these highly organization such that semantics inconsistencies are prevented. Meta category value hierarchies consist of:

- Meta category value
- Meta category value type
- Meta category value type class

6.2.2.1 Meta Category Value

A meta category value is a word or phrase that has abbreviations and a definition. Meta category values can exist within a hierarchy of meta category values. Meta category values exist within a type hierarchy. And finally, there are two classes of meta category value types, prefix and suffix. The rationale for meta category values et al is fully explained in the *Data Modeler's Architecture and Concept of Operations* book that is available from the Whitemarsh website.

It must be remembered that ONLY meta category values are the source for words or phrases that are either prefixed or suffixed to data element concepts, data elements, attributes, or columns. Meta category value hierarchies are employed only to distinguish between sets of meta category values. In general, Data element concepts have few if any assigned meta category values. That is because they are the most general and at the level of a data element *concept*. Then comes data elements, which are more precise, but still rather general. Finally comes context dependent uses, that is, attributes within entities and columns within tables. As the contextual level becomes more precise, there is a tendency to increase the quantity of meta category values.

Figure 46 presents the current set of meta category values within a highlighted meta category value type. The top browse shows the two meta category value type classes: prefix and suffix. For suffix, these are data use modifiers. And within that, there are, for example, qualitative modifiers and quantitative modifiers. Within quantitative modifiers there is money. Money then is the meta category value type that then becomes a meta category value, amount, and below amount are the different kinds of amount. That is, average, balance, and cost.

Meta category values can either be entered in a hierarchy or as siblings. If it's to be a sibling then highlight the parent meta category value and then press enter. The screen in Figure 47 is presented. Enter the name, abbreviations and description. In this case, what ever is entered will be in a sibling relationship with text, description, date, and definition.

If an additional level of hierarchy is desired below a meta category value then highlight the meta category value and then press enter. Enter the information as required by Figure 47 and then press enter. The new meta category value and level will then appear.

Creating all these meta category value types and values is not simple. It takes much reflection and care. What is accomplished, however is the complete taxonomic classification of all the words that, in collections, define the semantics of all enterprise persistent data. Well done, data becomes semantically interoperable. The alternative is not just semantically unacceptable, it

is very wasteful of human and financial resources. Semantic chaos costs many times more than semantic harmony.

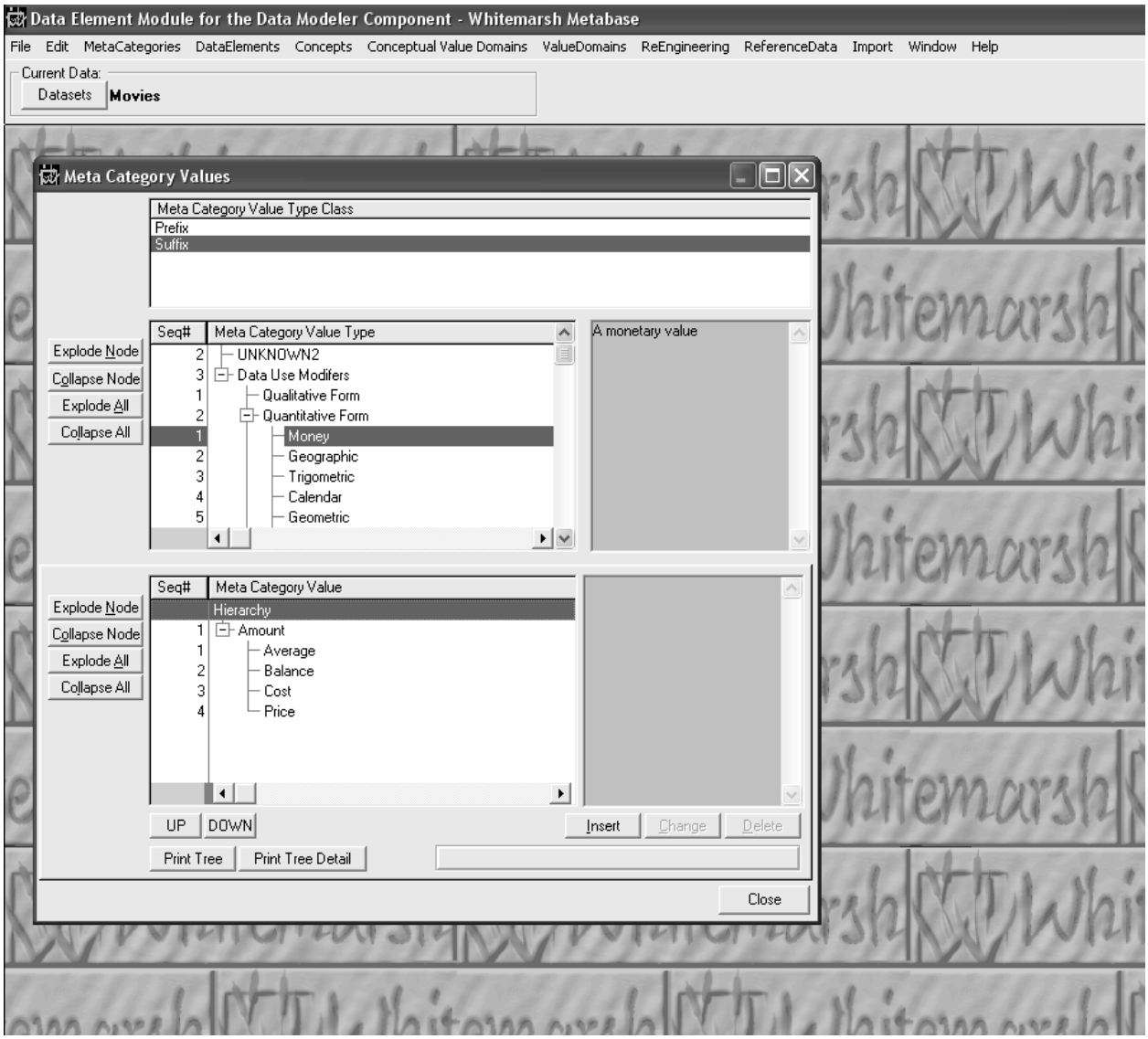

Figure 46. Meta Category Values.

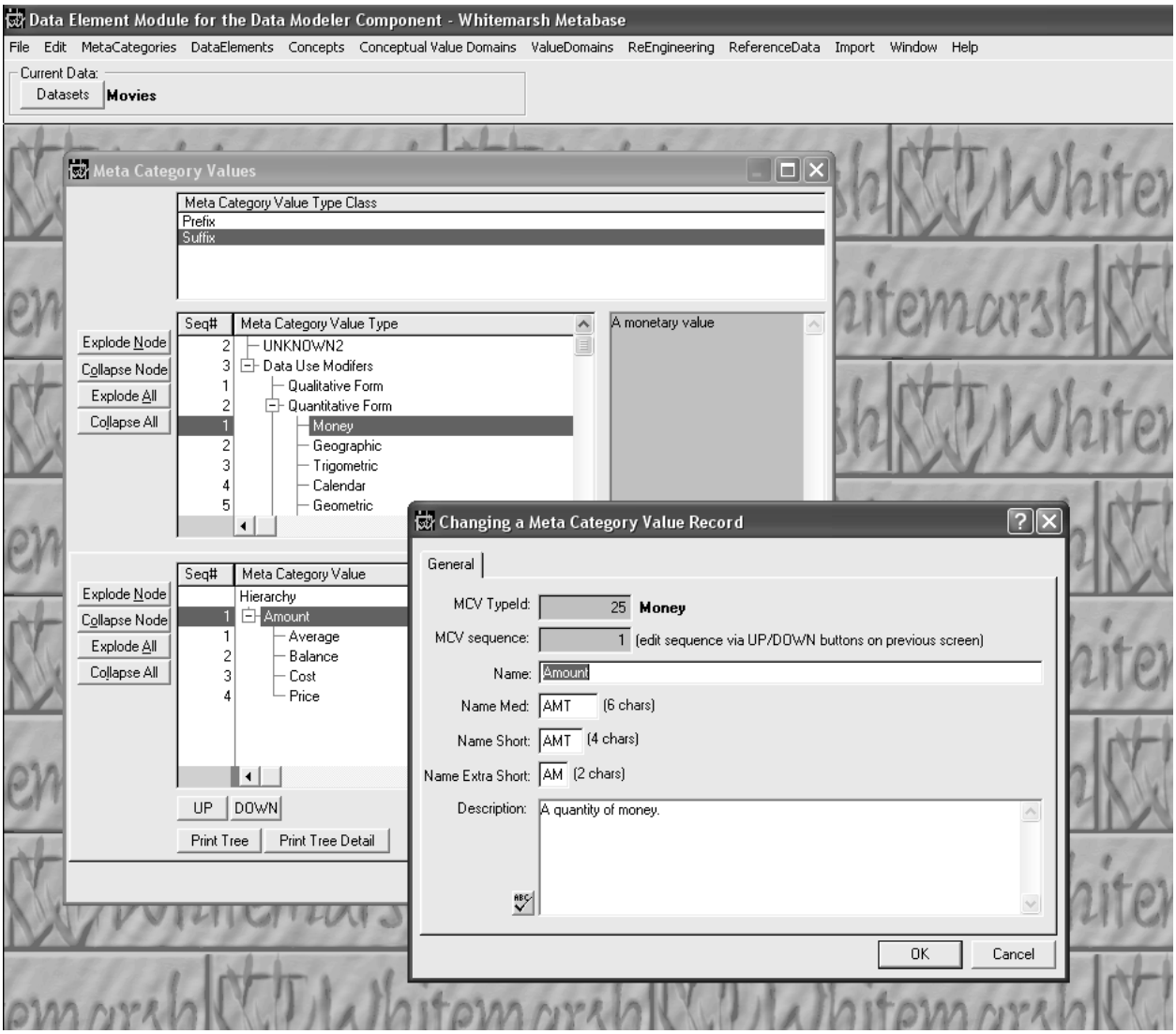

Figure 47. Meta Category Value update screen.

6.2.2.2 Meta Category Value Type

Meta category value types are a type of classification hierarchy. Meta category value hierarchies exist within specific meta category value type hierarchies. Figure 48 presents a list of meta category value types. From this window there are two super-classes, prefix and suffix. Show are is the hierarchy for prefix, that is, semantic modifiers. There may be additional prefix trees. Within this tree the three shown are Accuracy, Geographic, and Temporal. The highlighted meta category value type accuracy in turn causes the current list of meta category values to be displayed. In Figure 48, the Sequence Number column shows that Accuracy meta category value word choices will appear before geographic category value word choices before temporal meta category value word choices. If the metabase administrator wishes to change that sequence then the Up|Down buttons can be used.

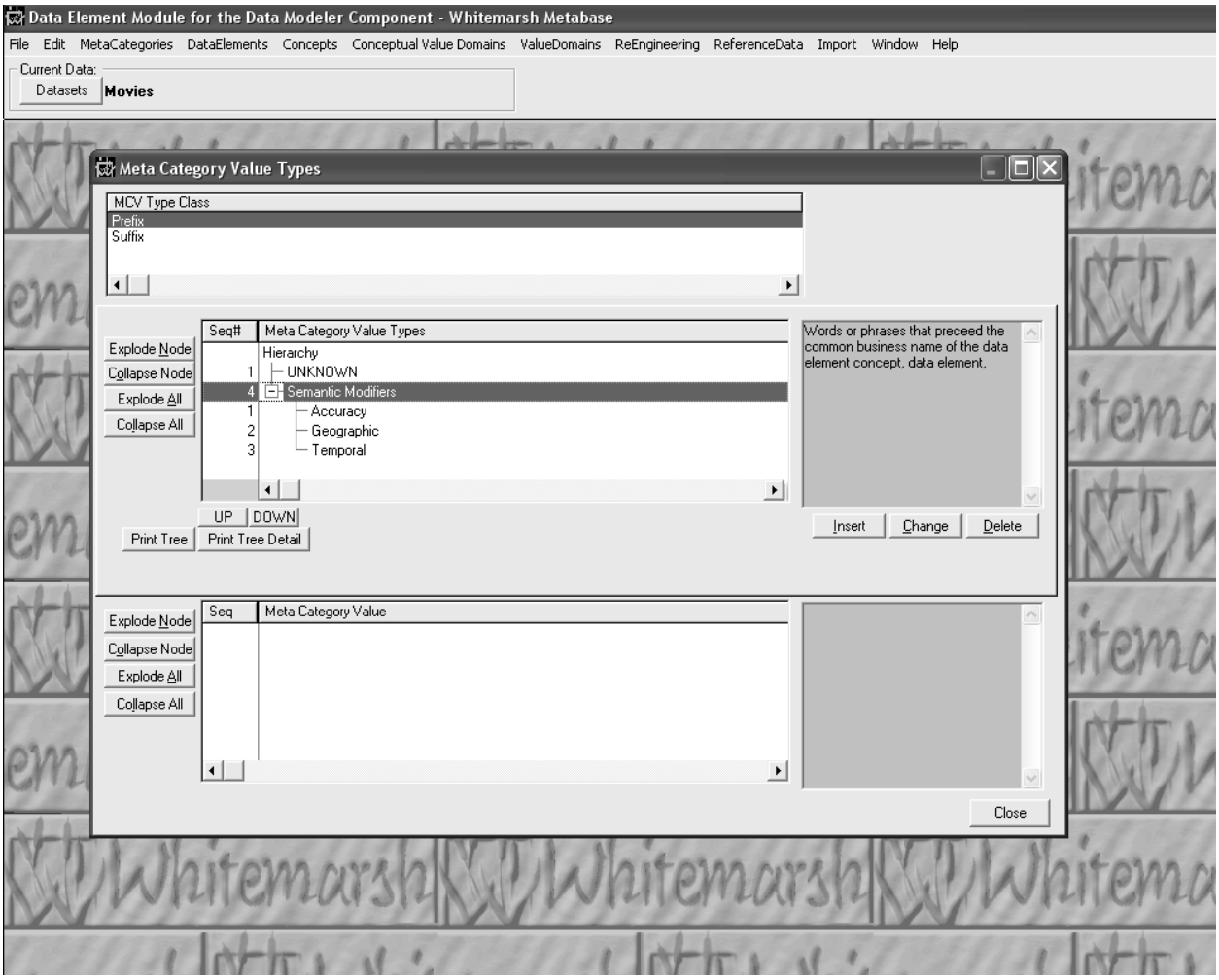

Figure 48. Meta Category Value Types.

Meta category value types can either be entered in a hierarchy or as siblings. If it's to be a sibling then highlight the parent meta category value type and then press Insert. The screen in Figure 49 is presented. Enter the name, abbreviations and description. In this case, what ever is entered will be in a sibling relationship with text, description, date, and definition.

If an additional level of hierarchy is desired below a meta category value type then highlight the meta category value type and then press Insert. Enter the information as required by Figure 49 and then press enter. The new meta category value type and level will then appear. If a meta category value type already has assigned meta category values then a lower level new meta category value type cannot be created.

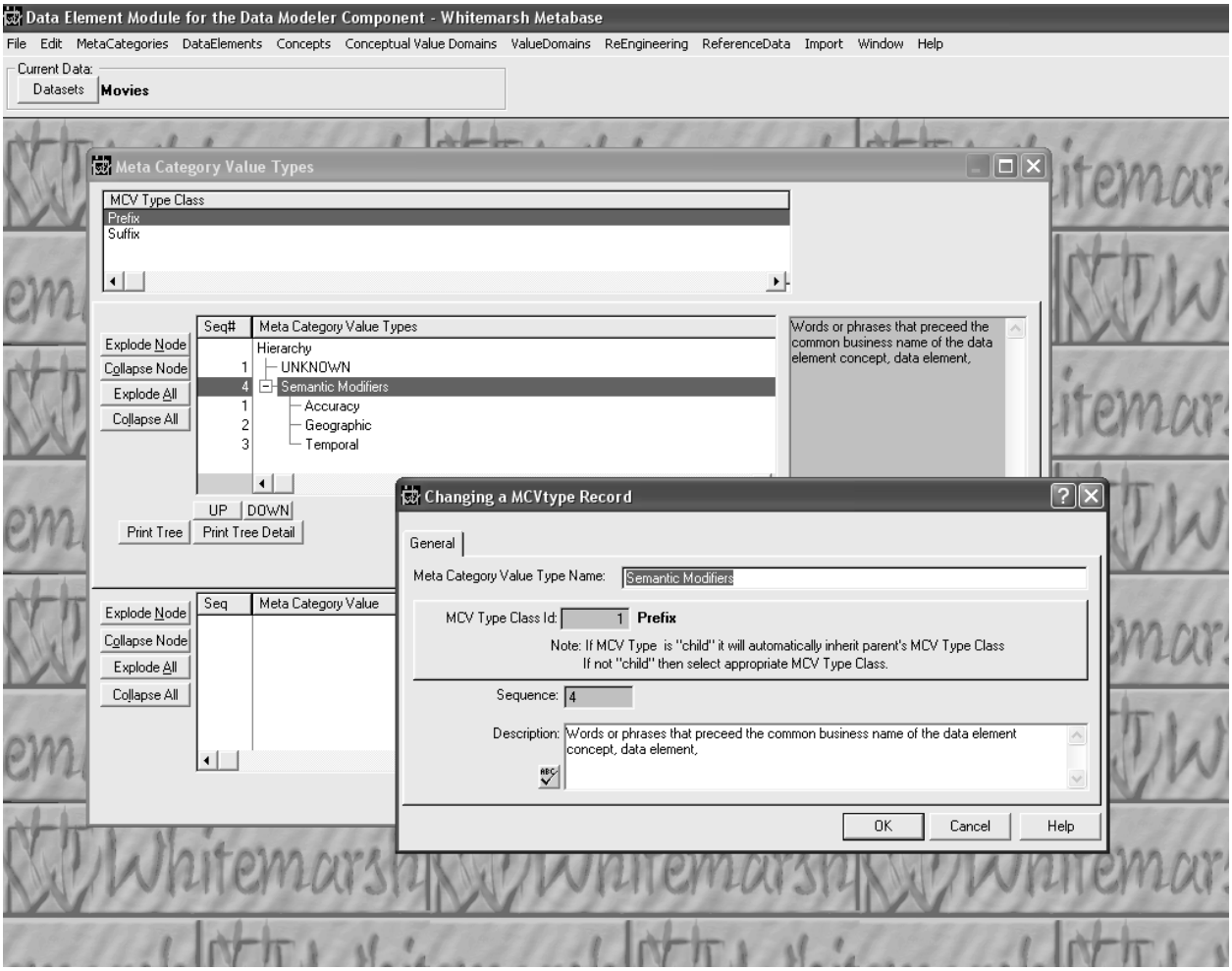

Figure 49. Meta Category Value Type update screen.

6.2.2.3 Meta Category Value Type Class

Figure 50 shows the two instances of meta category value type classes. The only two allowed values are prefix and suffix. All meta category value types and contained meta category values are thus grouped to appear either before the common business name or after the common business name. Figure 51 shows the update screen where the actual words that reflect prefix or suffix can be changed.

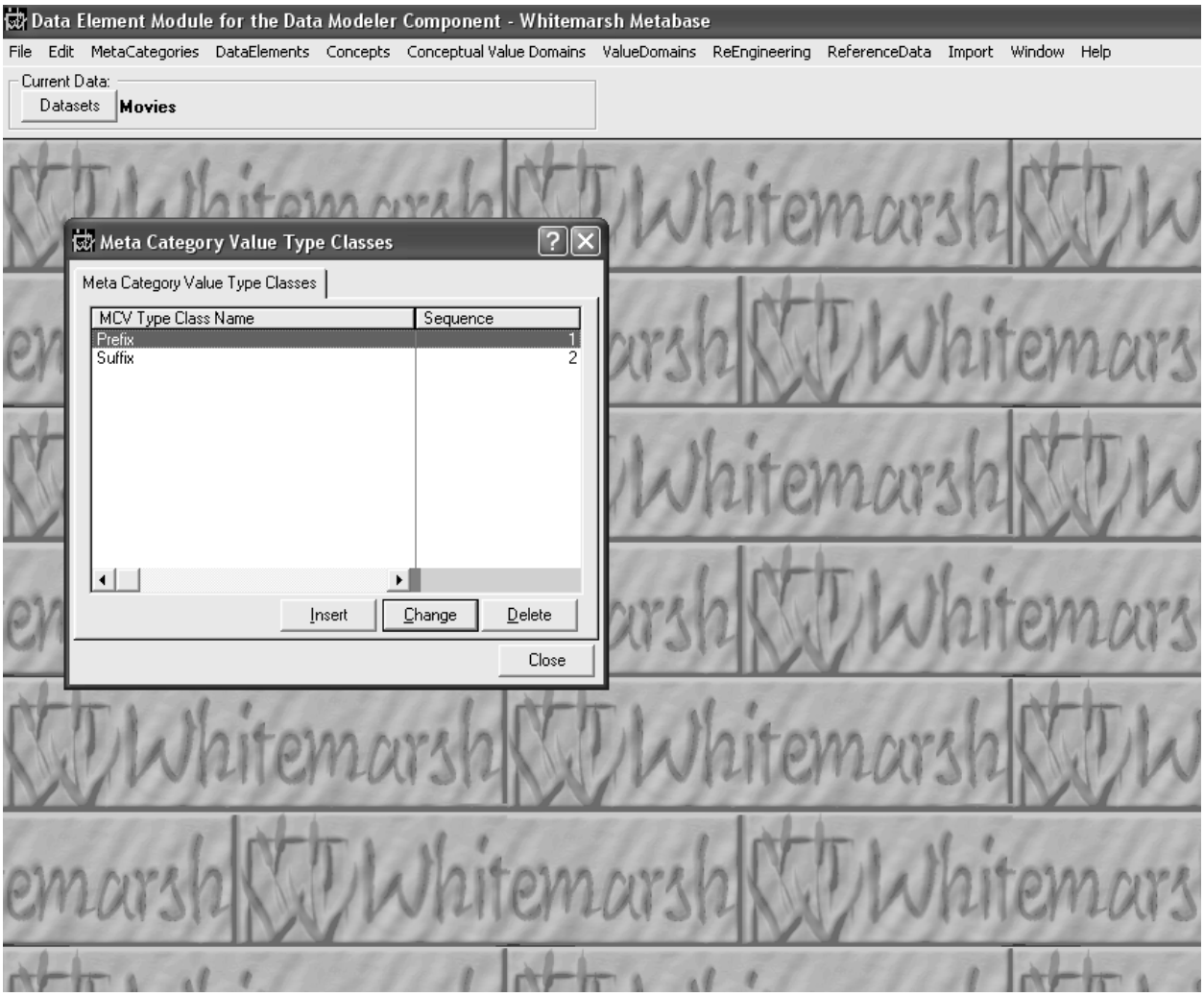

Figure 50. Meta Category Value Type Class.

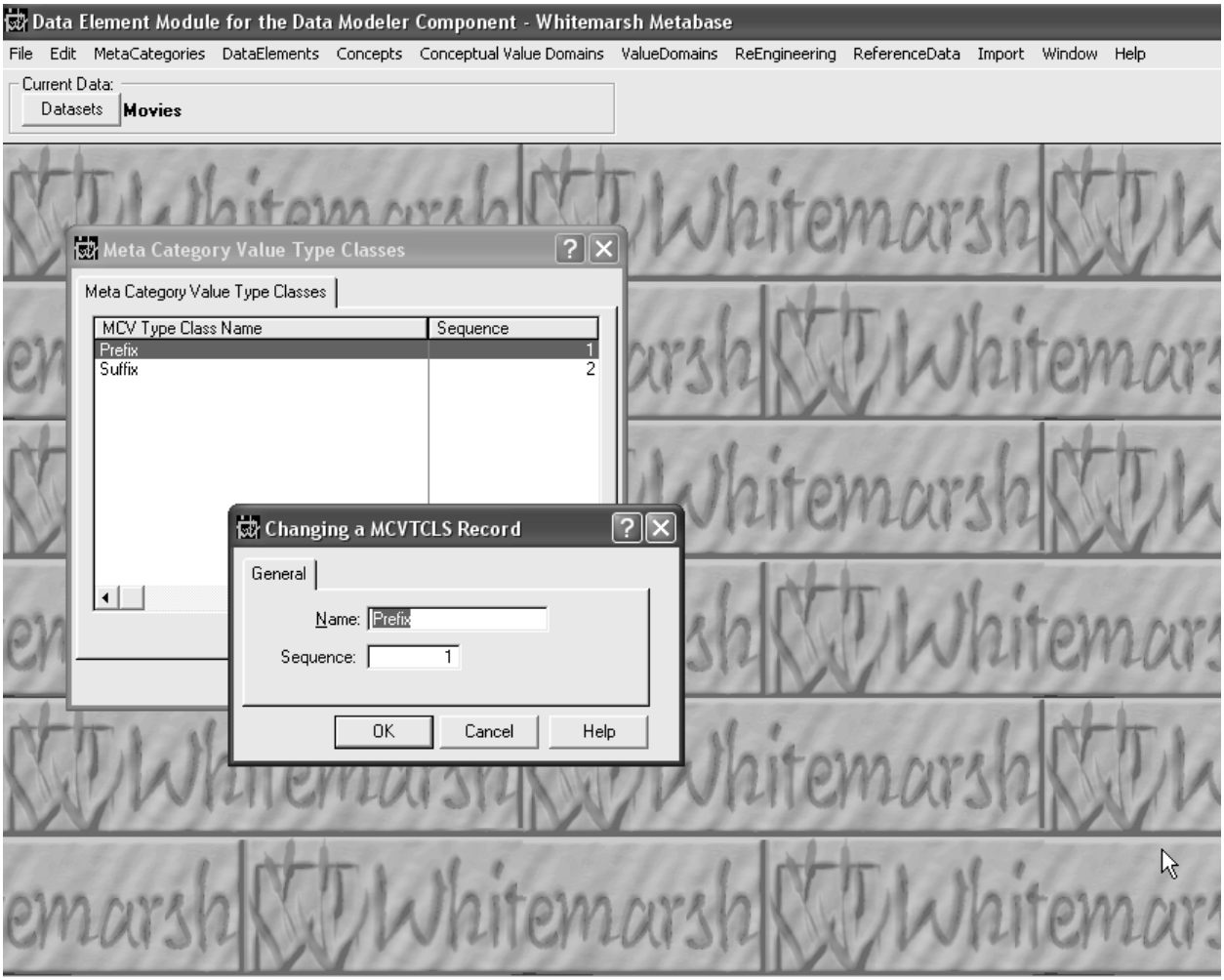

Figure 51. Meta Category Value Type Class update screen.

6.2.3 Concepts

Concepts, called Object Classes in the 11179 draft standard, are "a set of ideas, abstractions, or things in the real world that are identified with explicit boundaries and meaning and whose properties and behavior follow the same rules." They are called Concepts in the Whitemarsh metabase because it seems that this the best term for its definition. Concepts within the data element module of the Whitemarsh metabase are a basis for data element concepts. The other basis is conceptual value domains.

Concepts are constructed in a bill-of-materials fashion as are some other data element meta entity clusters. Consequently, there are

- Concepts
- Concept structures
- Concept structure types
- Concepts, Data Element Concepts and Data Elements

6.2.3.1 Concepts

A concept is a mechanism through which one or more data element concepts is classified. Figure 52 presents a list of concepts. Figure 53 presents the data update form for adding, deleting or changing a Concept.

6.2.3.2 Concept Structures

Concepts can exist singly or in hierarchies, or networks. In the last case, concepts form a traditional bill of materials data structure. Users are encouraged to view the Whitemarsh application WisBOM, Whitemarsh Bill of Materials. This application can be downloaded from the Whitemarsh website and it contains it's own data examples, supporting diagrams and documentation. This application is provided as a training device.

Figure 54 shows a existing set of concept structures. In this example, the concept, Asset contains two other concepts, Abstract Asset and Real Asset. Hence, the bill of materials.

When a concept name is highlighted and the Insert button is pressed, a screen like Figure 55 is presented. The specific concept that is desired as the contained concept is highlighted and then the select button is pressed. In this example, the concept, Event is being added to Asset.

There are three cases that are automatically prevented: twins, recursion and infinite recursion.

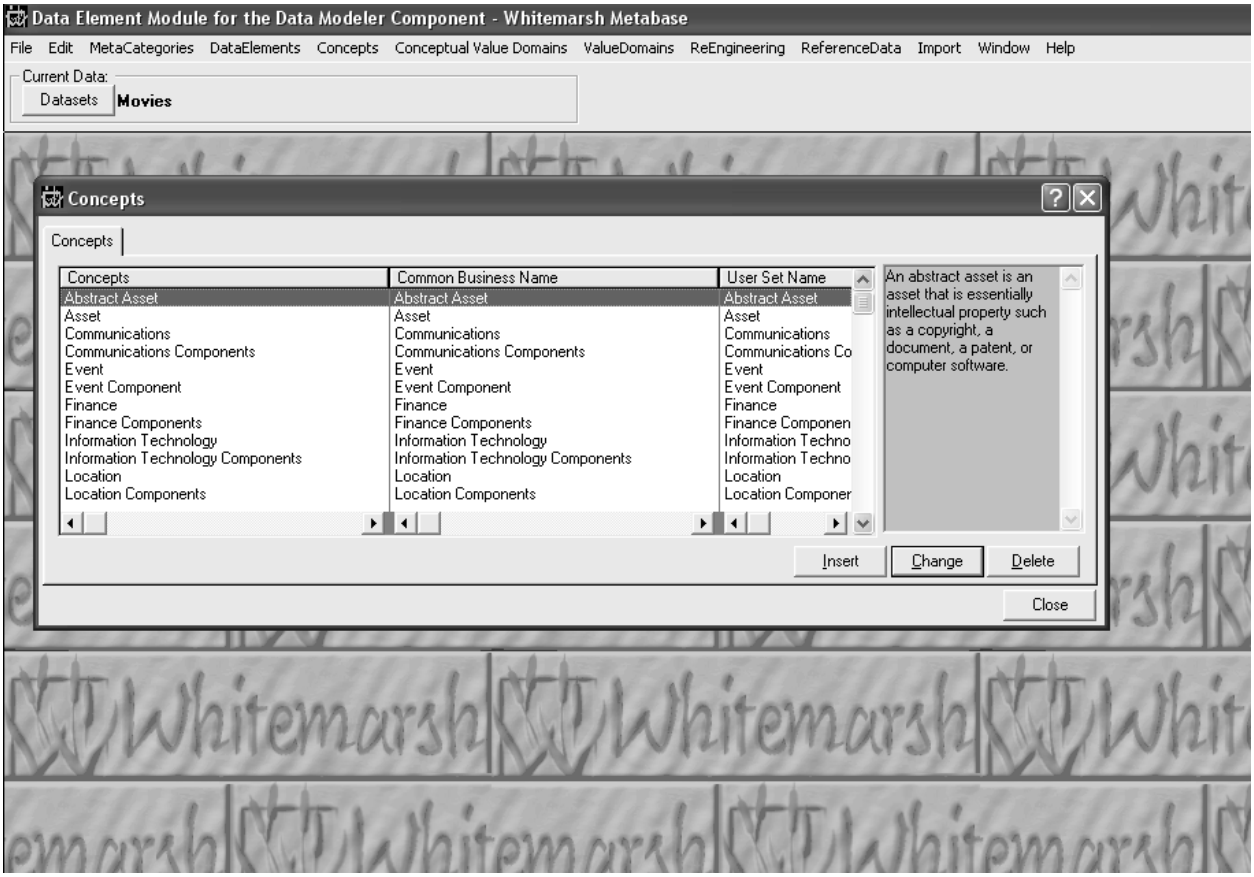

Figure 52. Concepts.

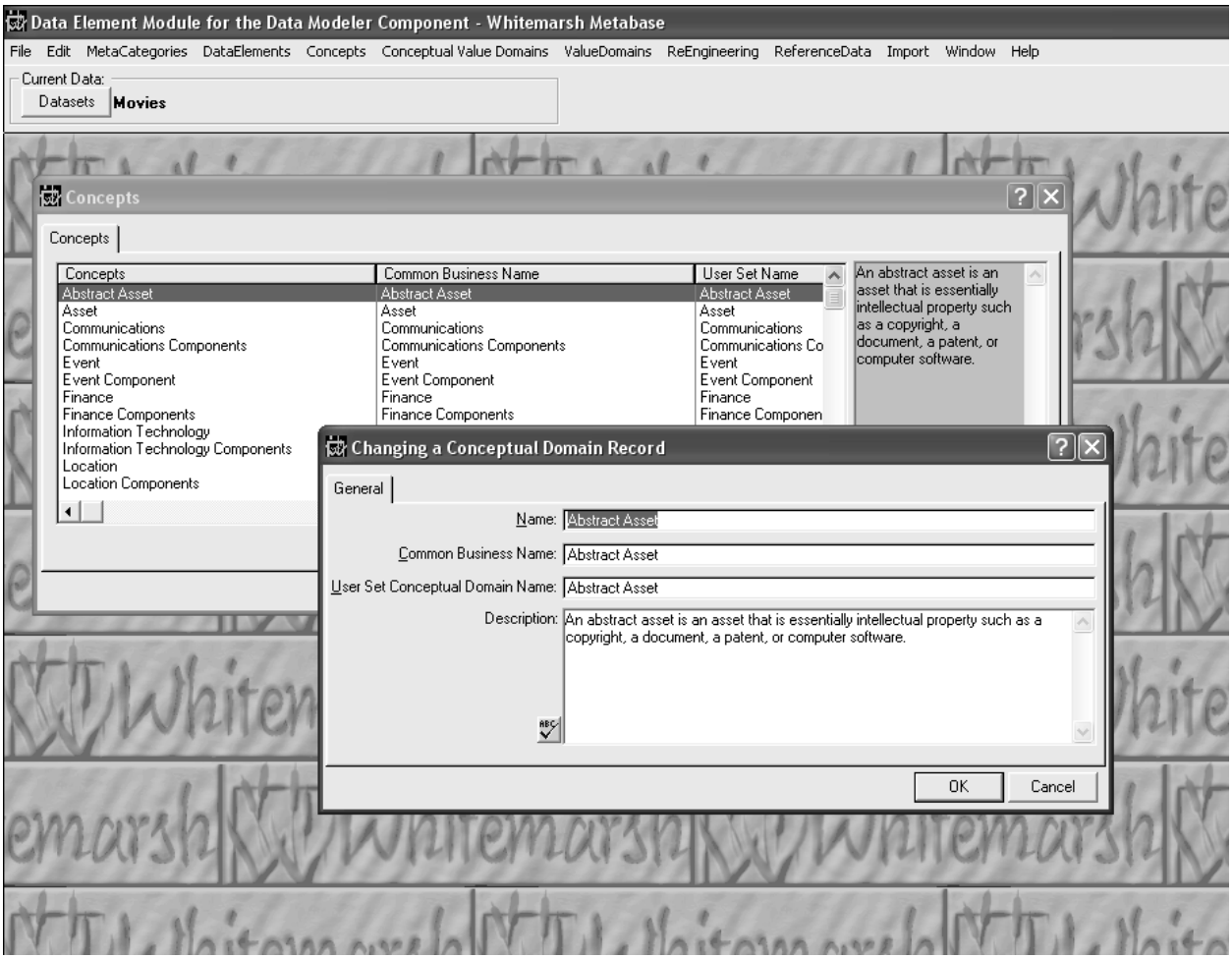

Figure 53. Concept update screen.

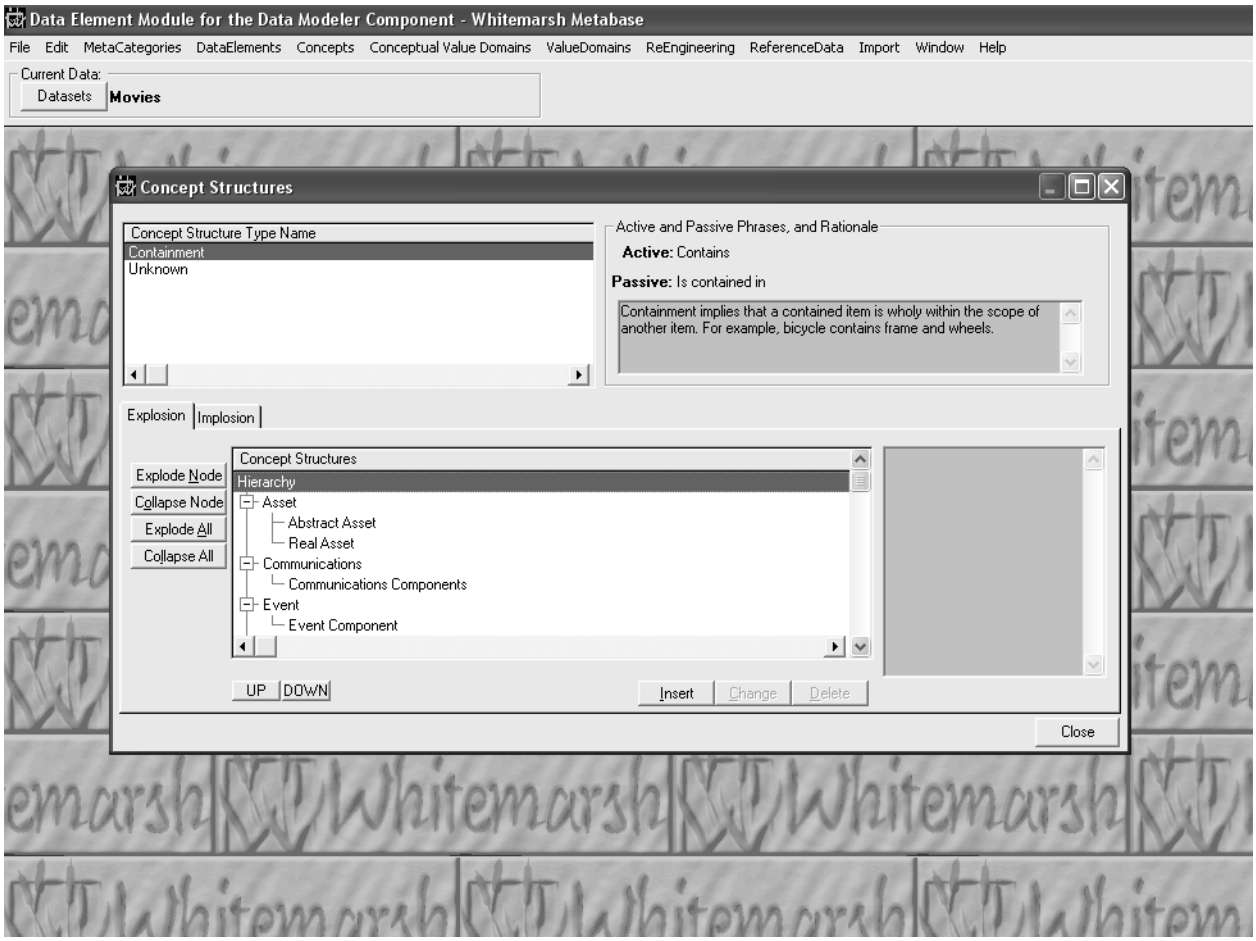

Figure 54. Concept Structures.

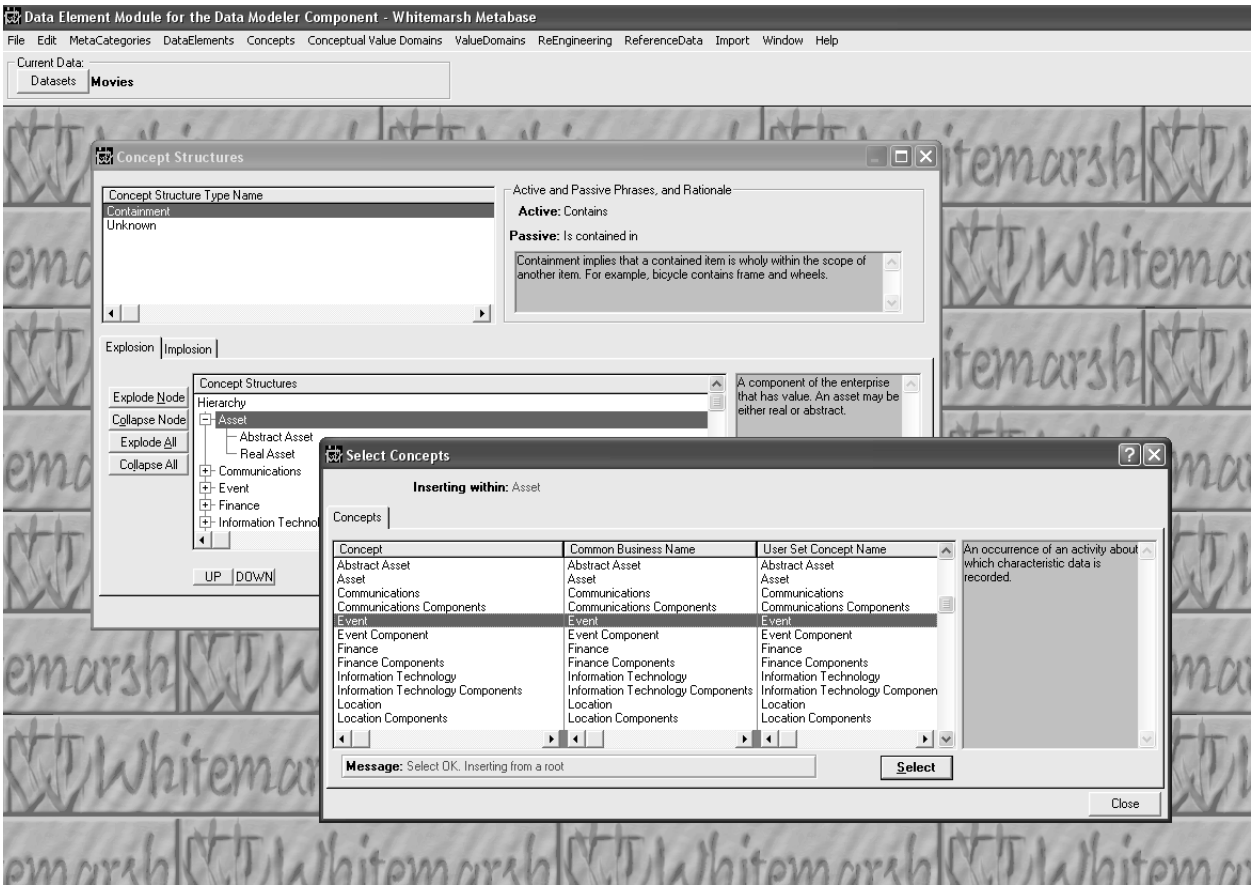

Figure 55. Concept Structure update screen.

6.2.3.3 Concept Structure Types

The concept structure type is a way of distinguishing one classification structure from another. Figure 56 presents the current list. If two different classification hierarchies are interconnected, it may be that the intersection is distinguished from the others by means of a different concept structure type.

Figure 57 presents the concept structure type update form. Not only is the name and description of the concept structure type provided, so too is the active phrase and the passive phrase. The active phrase is employed by the Whitemarsh metabase system when a down-ward structure is presented. For example, <parent> contains <child 1>, <child 2>, ..., <child n>. The passive phrase enables the reverse phrases to be presented. That is, <child 2> is contained in <parent>.

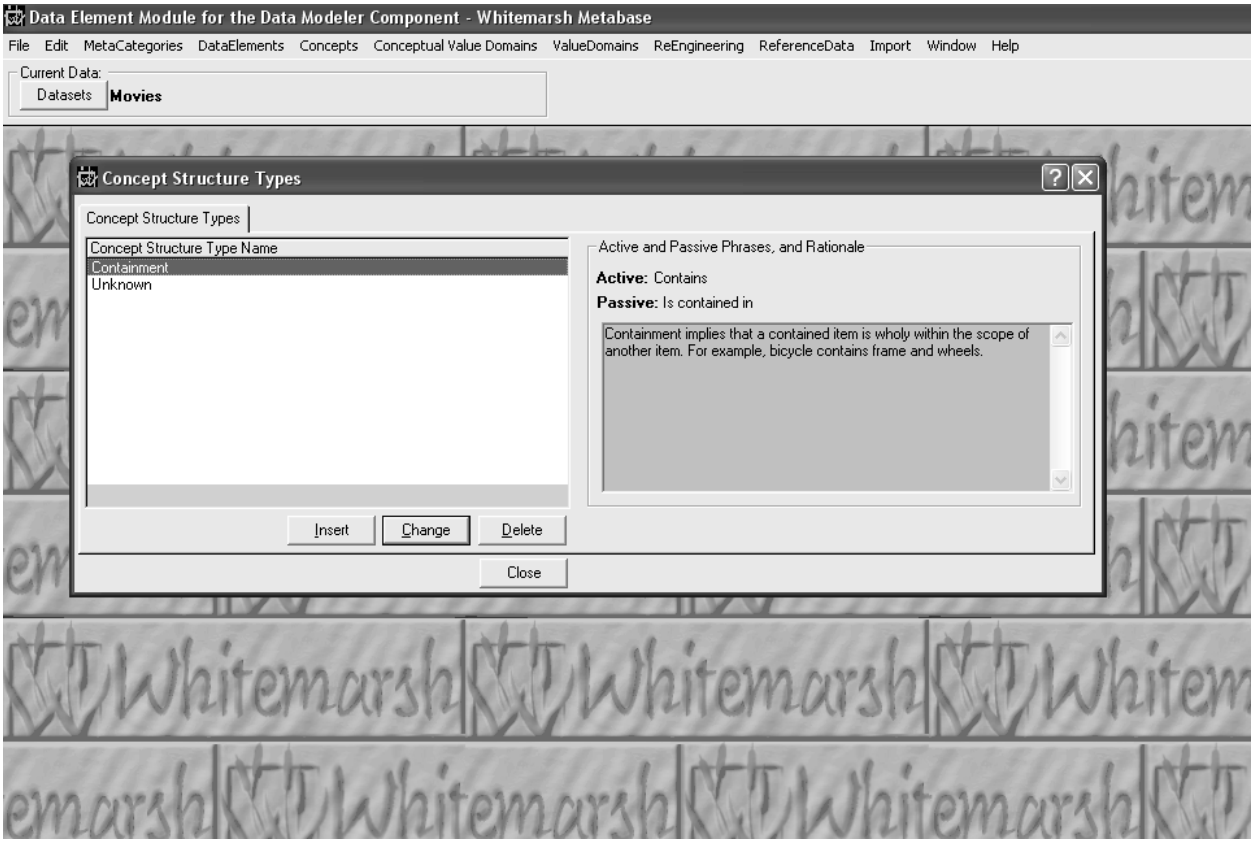

Figure 56. Concept Structure Types.

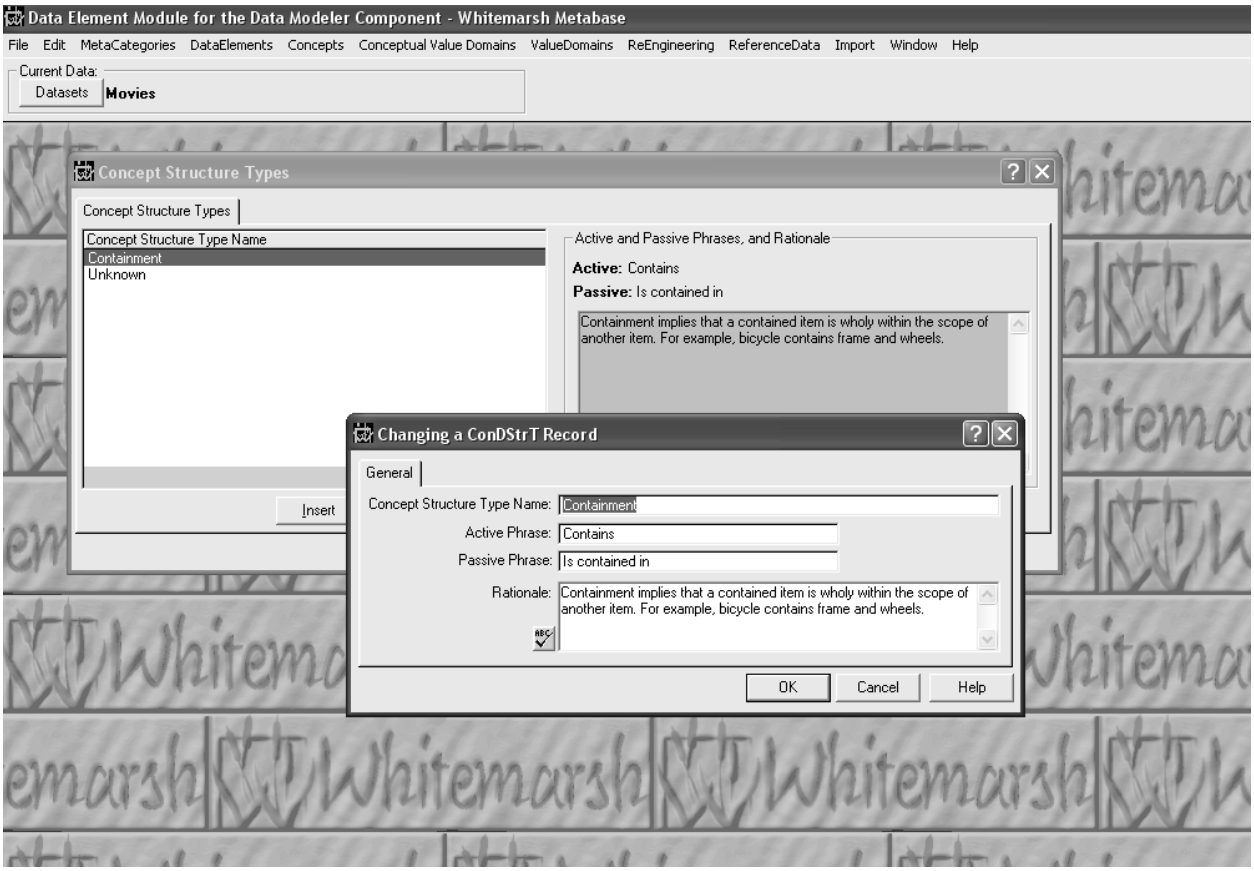

Figure 57. Concept Structure Type update screen.

6.2.3.4 Concepts, Data Element Concepts and Data Elements

Figure 58 presents concepts their associated data element concepts and data elements within their context of concepts and data element concepts. To arrive at the set of data elements, proceed left to right and top-down. Highlight and select the concept. Then, highlight and select the a particular data element concept. At this point the associated data elements are listed.

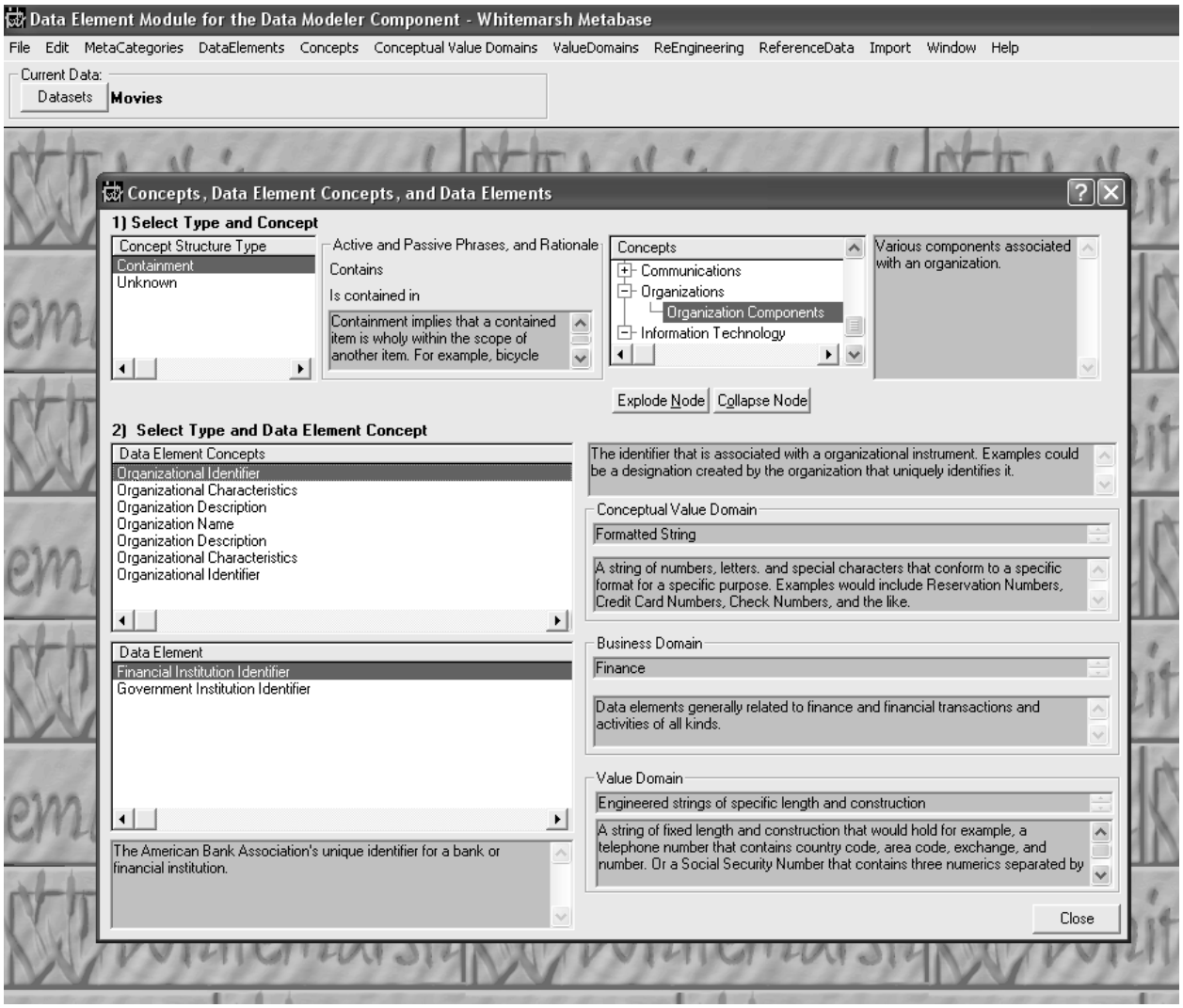

Figure 58. Concepts, Data Element Concepts, and Data Elements.

6.2.4 Conceptual Value Domains

Conceptual value domains are constructed in a bill-of-materials fashion as are some other data element meta entity clusters. Consequently, there are

- Conceptual value domains
- Conceptual value domain structures
- Conceptual value domain structure types

6.2.4.1 Conceptual Value Domains

In the ISO Standard 11179, Conceptual Value Domains are called Conceptual Domains. In the Whitemarsh metabase these are called Conceptual Value Domains because they are all about the concepts supporting Value Domains. Conceptual Value Domains are thus an overarching collection of conceptual meanings of value sets that are then specified in value domains. Proceeding from concepts are value domains. Value domains are then the start of nested value domains for data element concepts then data elements then attributes and then for columns. Each is intended to be a subset of the previous.

Figure 59 shows a current list of conceptual value domains. To add, change, or delete a conceptual value domain press the appropriate button. Figure 60 presents the update screen.

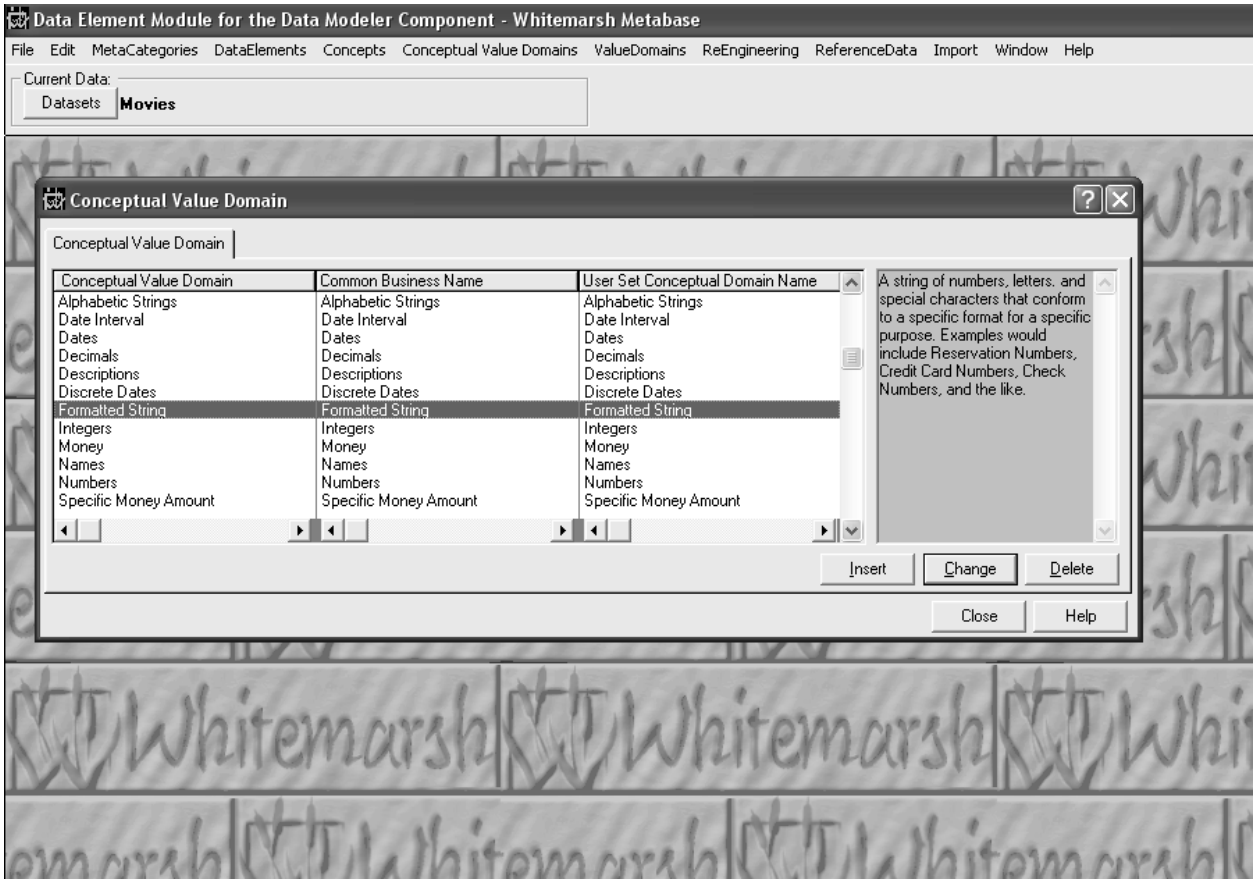

Figure 59. Conceptual Value Domains.

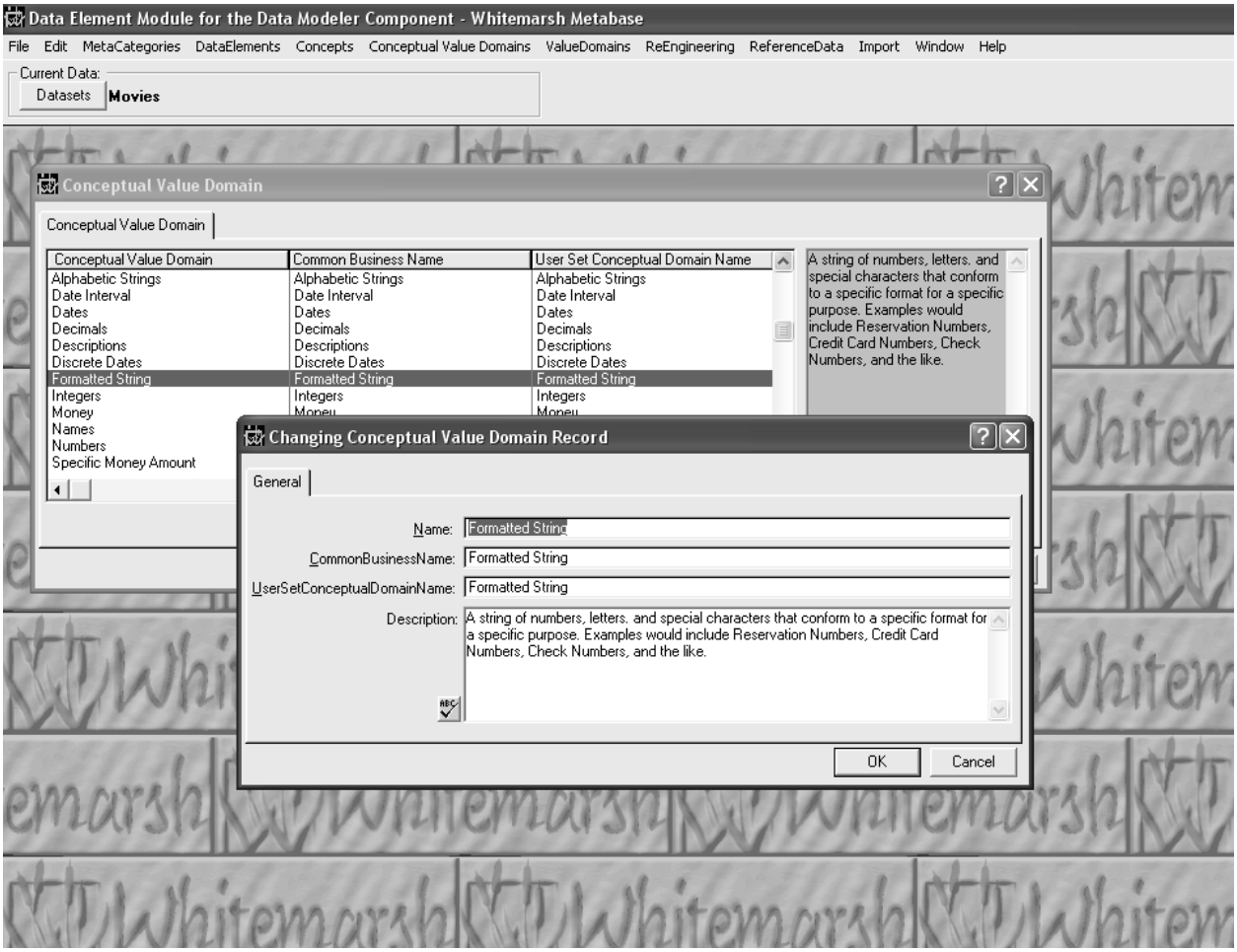

Figure 60. Conceptual Value Domain update screen.

6.2.4.2 Conceptual Value Domain Structures

Conceptual value domains can exist singly or in hierarchies, or networks. In the last case, conceptual value domains form a traditional bill of materials data structure. Users are encouraged to view the Whitemarsh application WisBOM, Whitemarsh Bill of Materials. This application can be downloaded from the Whitemarsh website and it contains it's own data examples, supporting diagrams and documentation. This application is provided as a training device.

Figure 61 shows a existing set of conceptual value domain structures. In this example, the same conceptual value domain is contained in two other conceptual value domains. Hence, the bill of materials.

When a conceptual value domain name is highlighted in the Conceptual Value Domain Structure browse and the Insert button is pressed, a screen like Figure 62 is presented. When the insert button was pressed, a screen appears that enable the selection of an additional conceptual value domain that is to be included in the existing set.

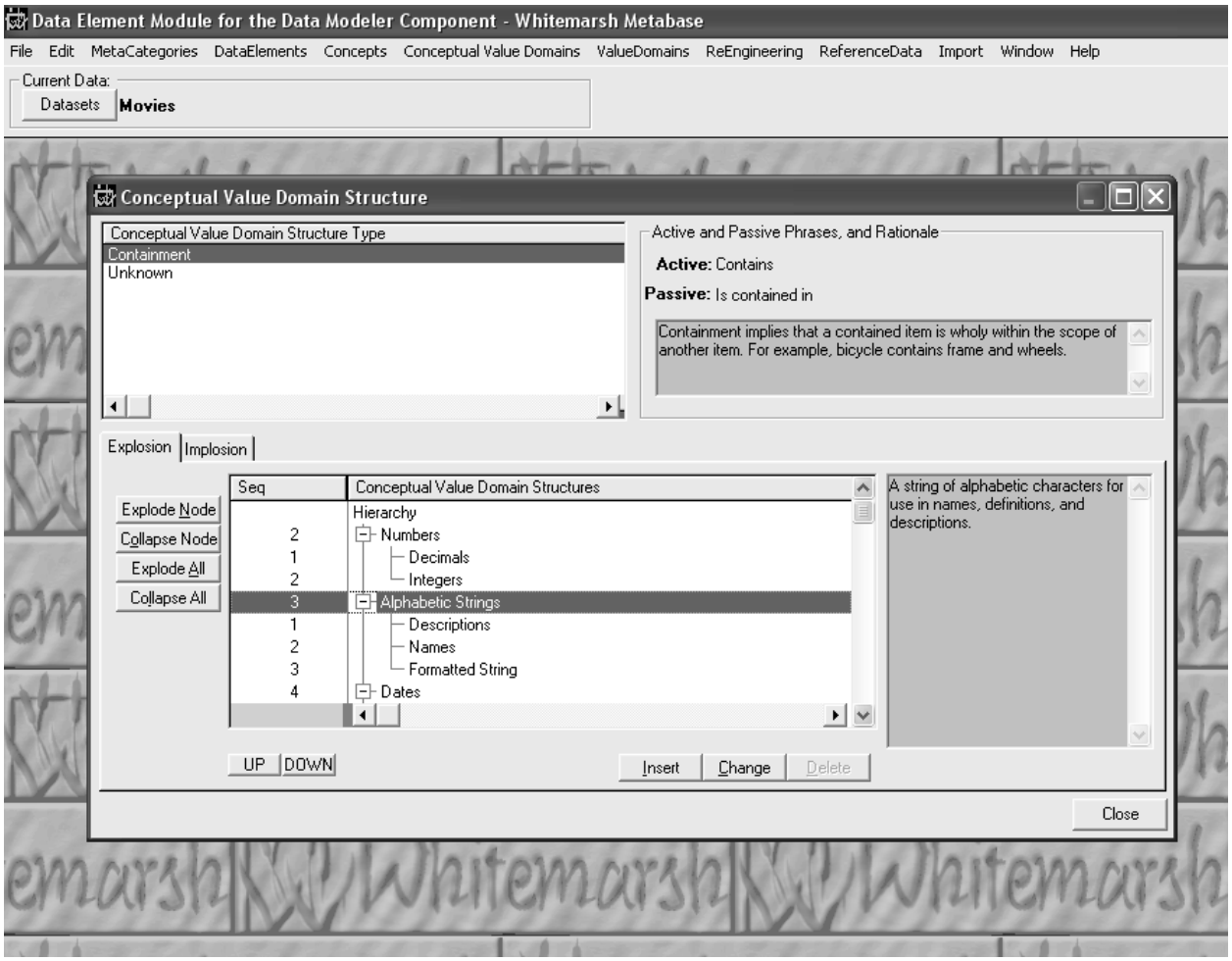

Figure 61. Conceptual Value Domain Structures.

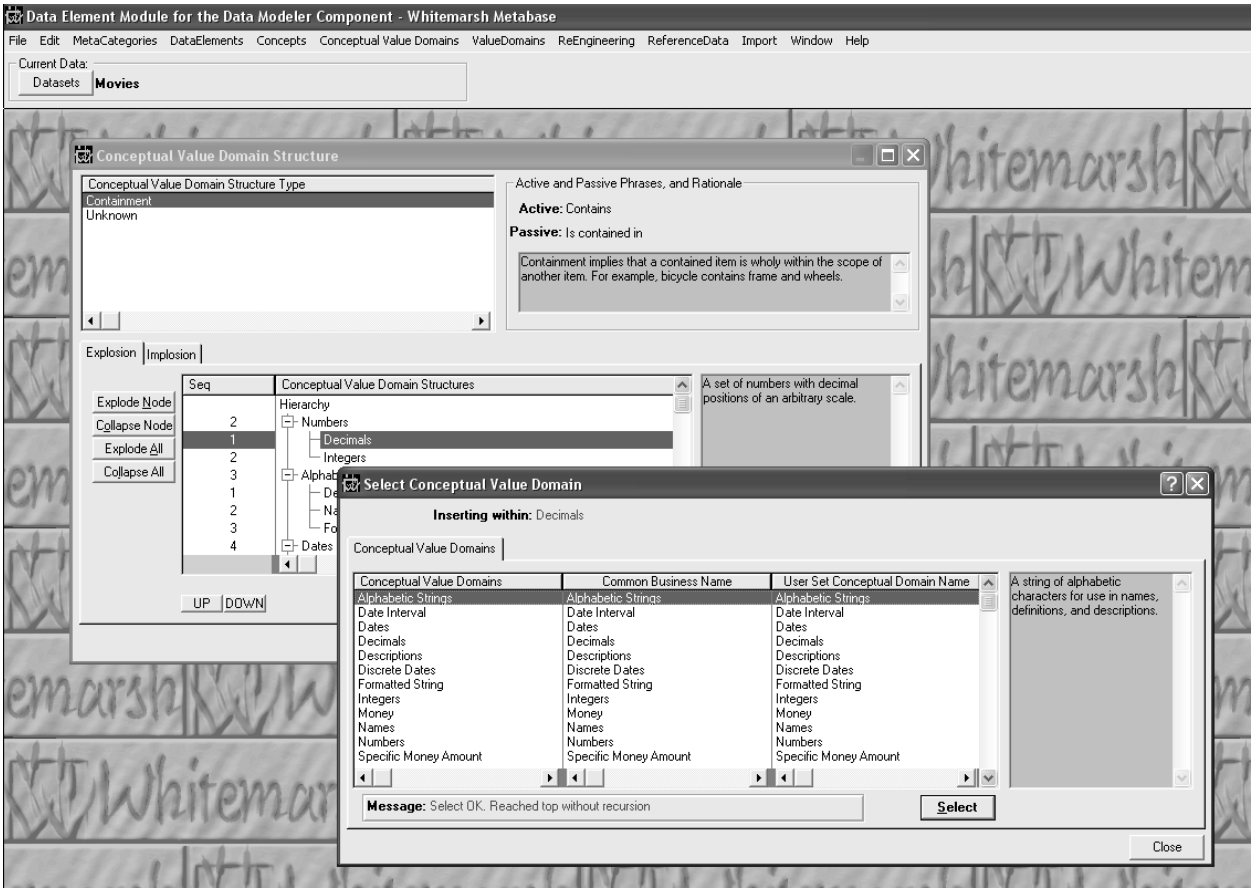

Figure 62. Conceptual Value Domain Structure update.

6.2.4.3 Conceptual Value Domain Structure Types

The conceptual value domain structure type is a way of distinguishing one conceptual value domain structure from another. Figure 63 presents the current list. If two different conceptual value domain structure hierarchies are interconnected, it may be that the intersection is distinguished from the others by means of a different conceptual value domain structure type.

Figure 64 presents the conceptual value domain structure type update form. Not only is the name and description of the conceptual value domain structure type provided, so too is the active phrase and the passive phrase. The active phrase is employed by the Whitemarsh metabase system when a down-ward structure is presented. For example, <parent> contains <child 1>, <child 2>, ..., <child n>. The passive phrase enables the reverse phrases to be presented. That is, <child 2> is contained in <parent>.

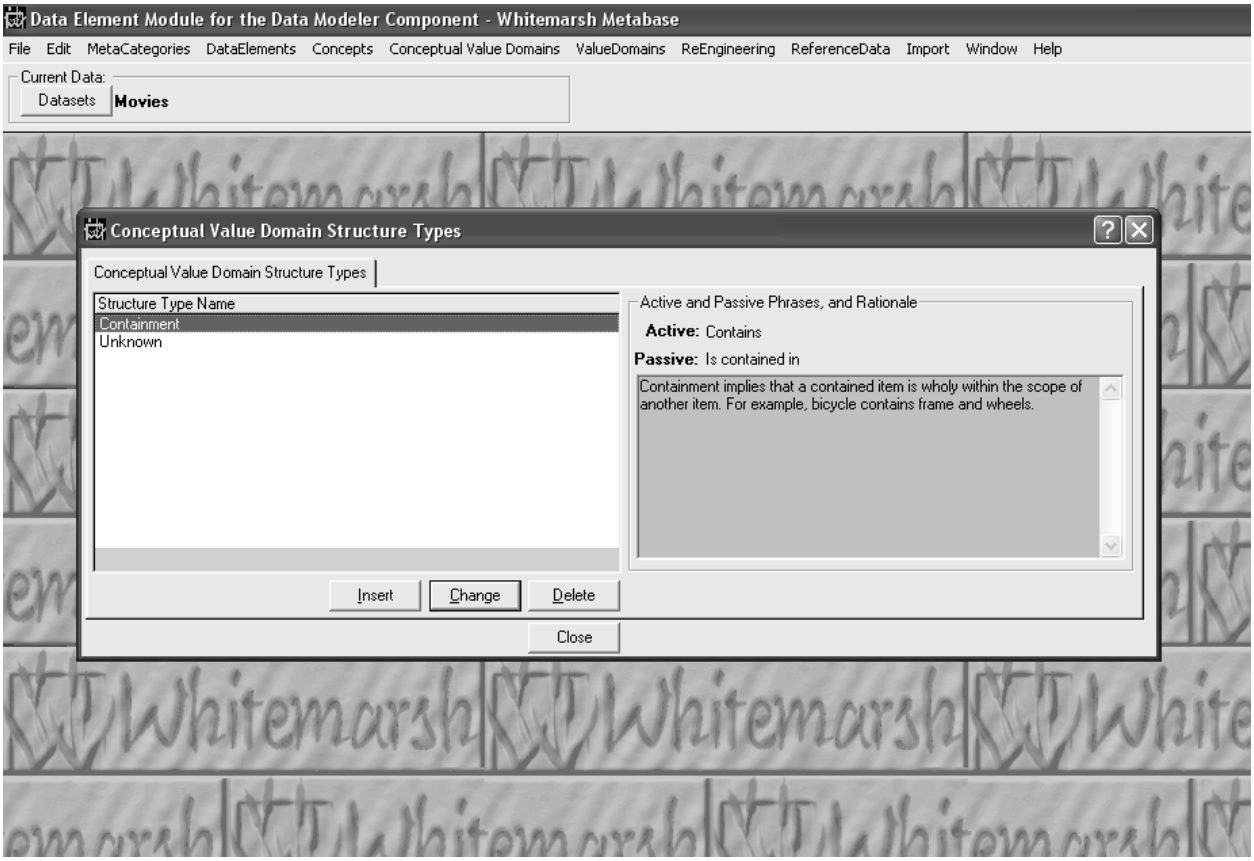

Figure 63. Conceptual Value Domain Structure Types.

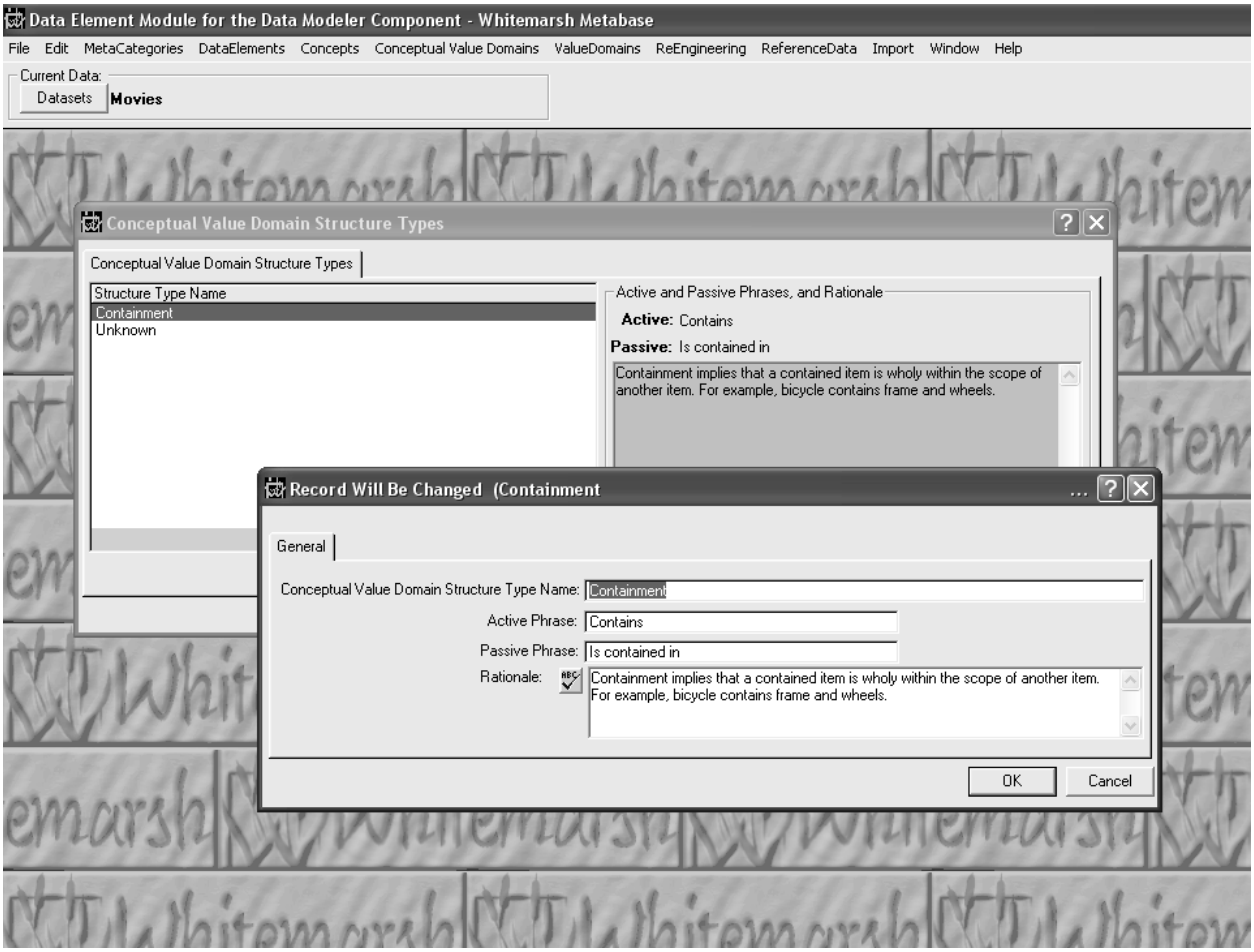

Figure 64. Conceptual Value Domain Structure Type update screen.

6.2.4.4 Conceptual Value Domains, Data Element Concepts, and Data Elements

Figure 65 presents conceptual value domain their associated data element concepts and data elements within their context of conceptual value domain and data element concepts. To arrive at the set of data elements, proceed left to right and top-down. Highlight and select the conceptual value domain. Then, highlight and select the a particular data element concept. At this point the associated data elements are listed.

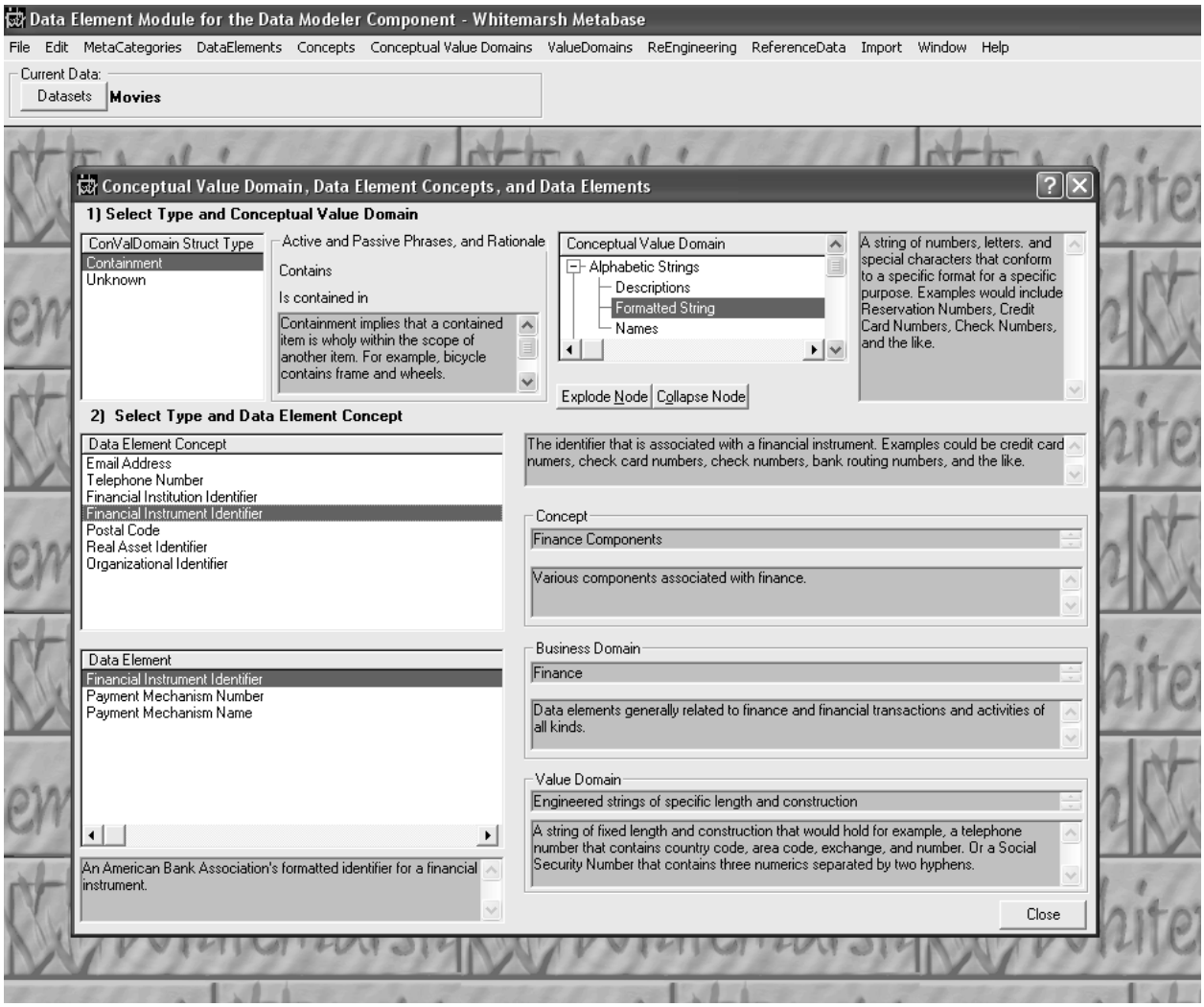

Figure 65. Conceptual Value Domain, Data Element Concepts, and Data Elements.

6.2.5 Value Domains

Value Domains are collections of values that may be employed to provide discrete value-bases representation for the existence of data elements as columns within tables of a database. In short, the concepts, and allowed value representations for the columns of data rows. Fully defined value domains consist of:

- Value Domains
- Value Domain Structures
- Value Domain Structure Types

6.2.5.1 Value Domains

Value domains are collections of values that exist within the context of a conceptual value domain. Specific value domains are assigned to one or more data elements. If a data element's assigned value domain is to apply to attributes, columns, or DBMS columns then nothing more need be done as the ultimate ancestor of an attribute, column, or DBMS column is the data element. If only a subset of values from within the data element's value domain are to be assigned to an attribute, column, or DBMS column then the subsets are defined with value domain structures and are assigned as appropriate to attributes, columns, or DBMS columns.

Value domains are constructed in a bill-of-materials fashion as are some other data element meta entity clusters. Consequently, there are

- Value domains
- Value domain structures
- Value domain structure types
- Value domain data types

The sets of permissible values are able to be constrained within value domains that are defined for a specific value domain.

If the set of value domains within the value domain value domain are sufficient then value domains for data element concepts, data elements, attributes and columns need not be entered. Conversely, if there is to be a more refined set within data element concept value domains, data elements, attributes or columns then they should be entered in those appropriate places. Figure 66 presents the current list of value domains. Figure 67 presents the update form for adding, changing or deleting value domains. If the selected data type supports quantity of decimal places then

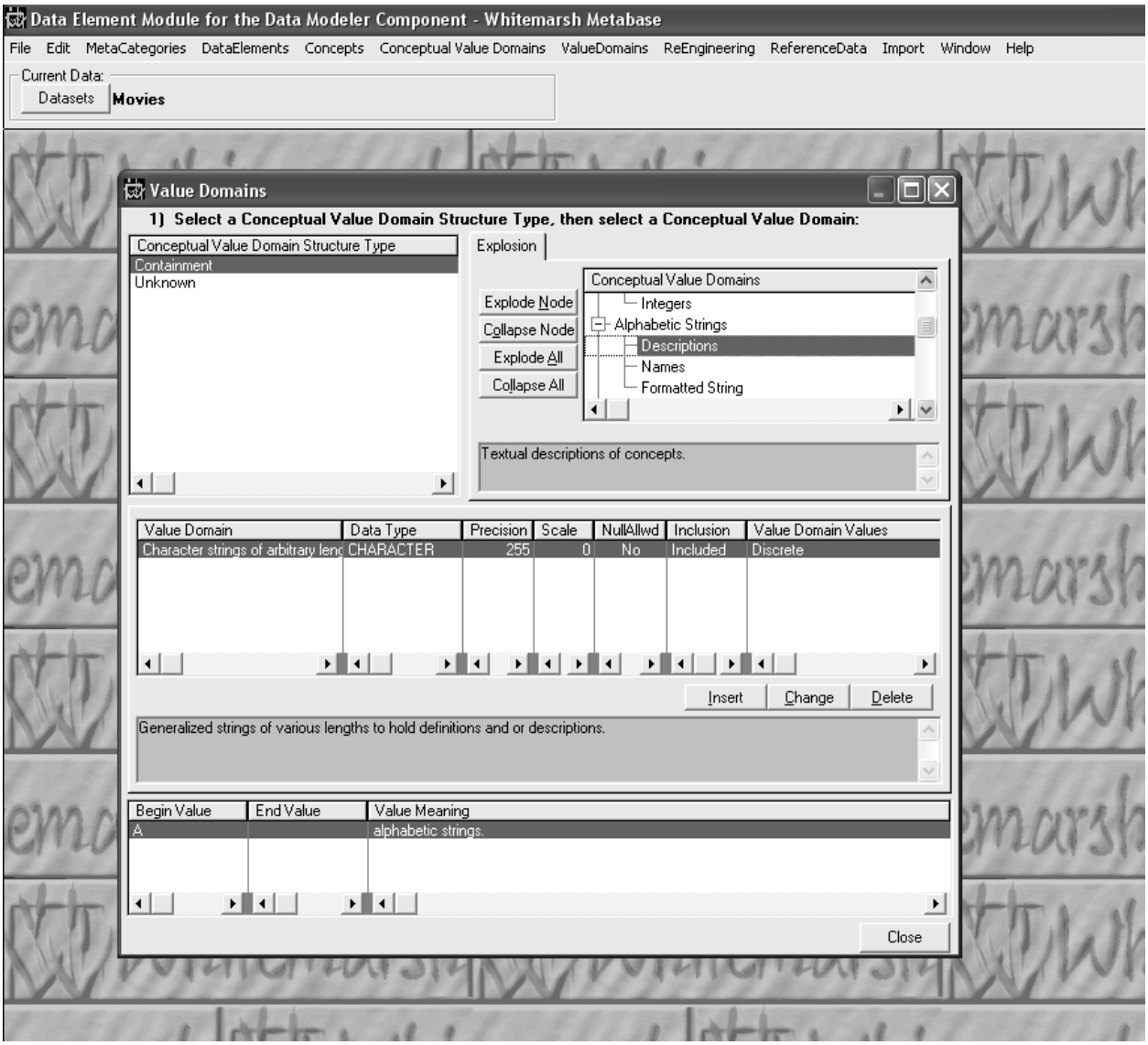

Figure 66. Value Domains.

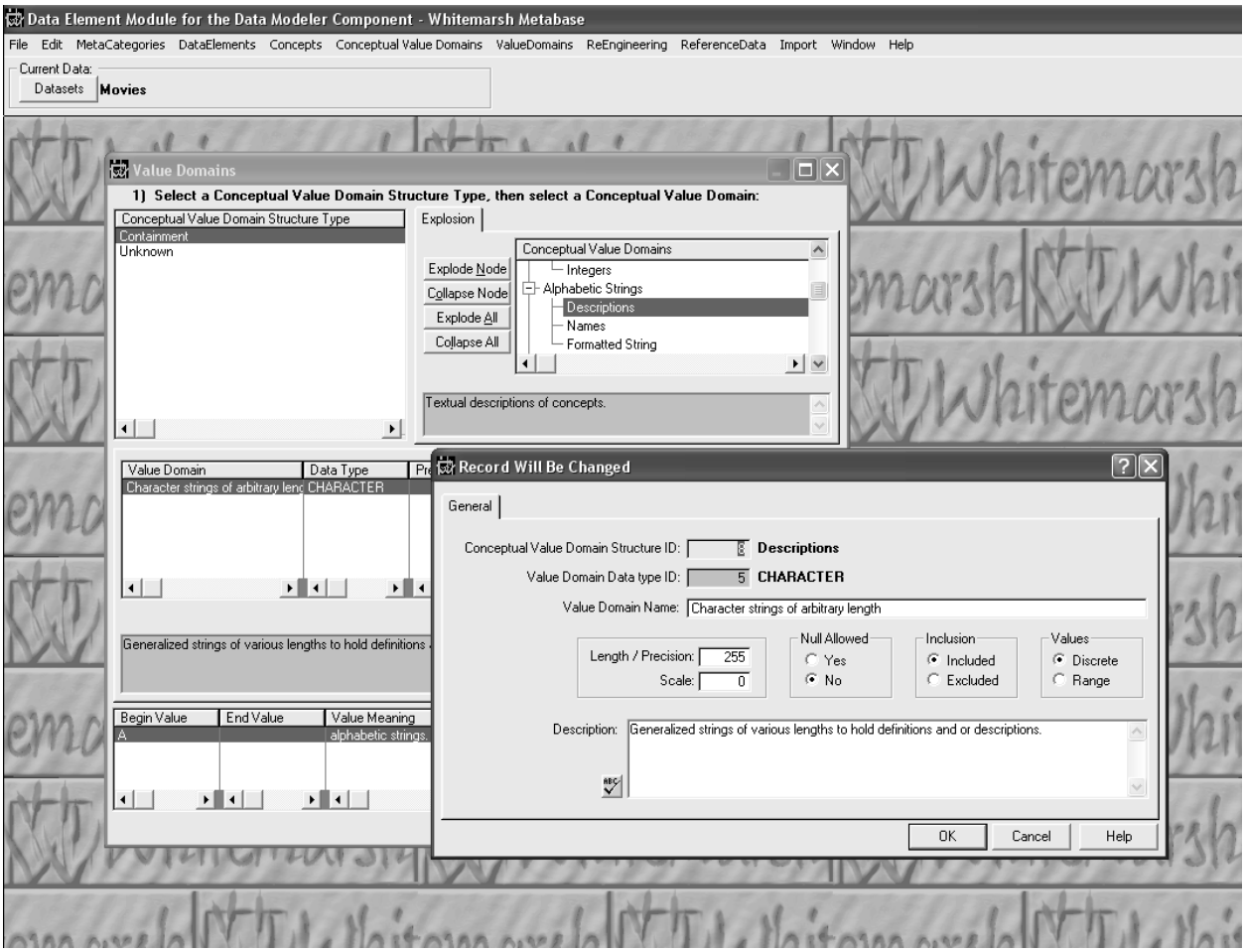

Figure 67. Value Domain update screen.

6.2.5.2 Value Domain Structures

Value domains can exist singly or in hierarchies, or networks. In the last case, value domains form a traditional bill of materials data structure. Users are encouraged to view the Whitemarsh application WisBOM, Whitemarsh Bill of Materials. This application can be downloaded from the Whitemarsh website and it contains it's own data examples, supporting diagrams and documentation. This application is provided as a training device.

Figure 68 shows a existing set of value domain structures. In this example, the same value domain is contained in two other value domains. Hence, the bill of materials.

When a value domain name is highlighted and the Insert button is pressed, a screen like Figure 69 is presented. When the insert button was pressed, a screen appears that enable the selection of an additional value domain that is to be included in the existing set.

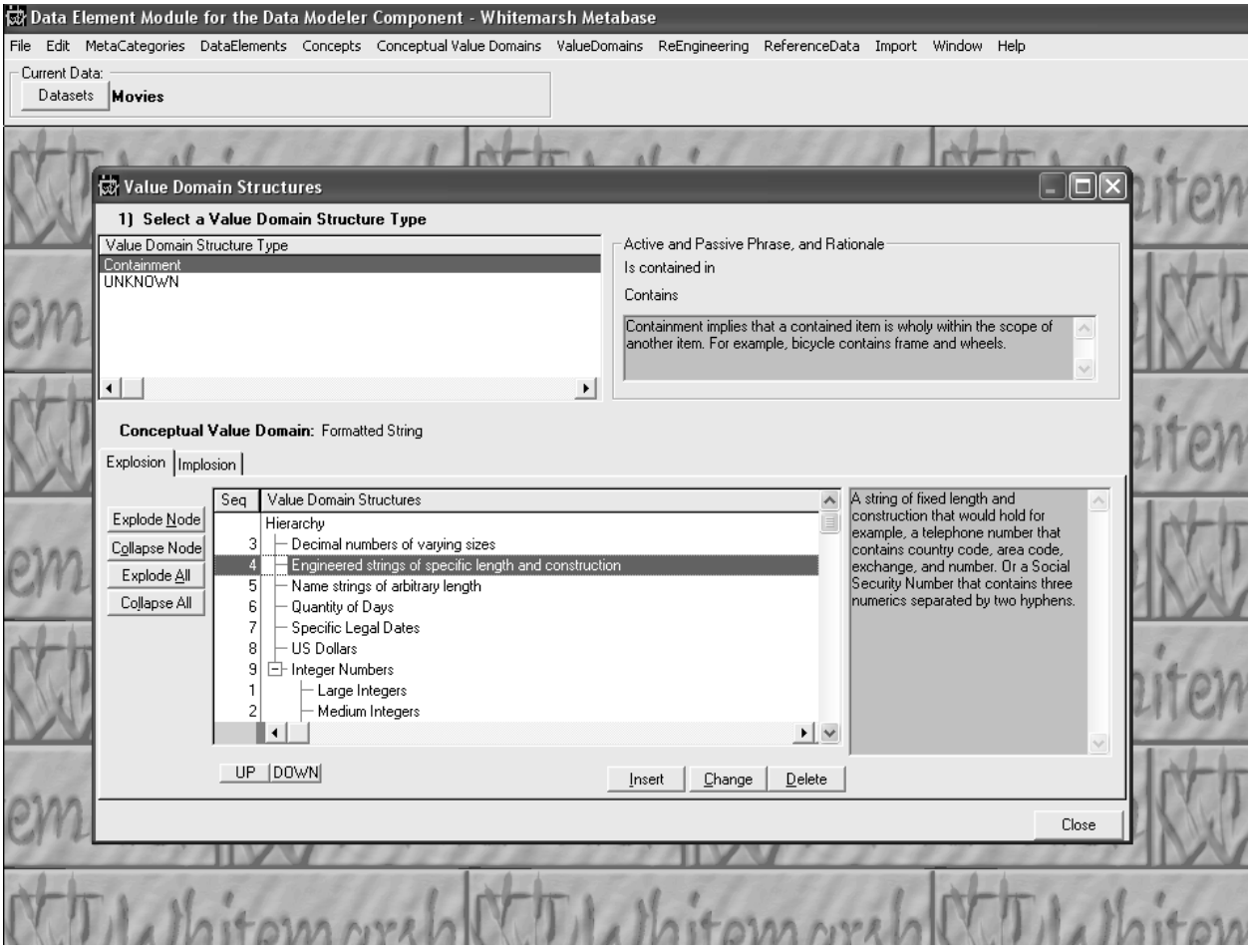

Figure 68. Value Domain Structures.

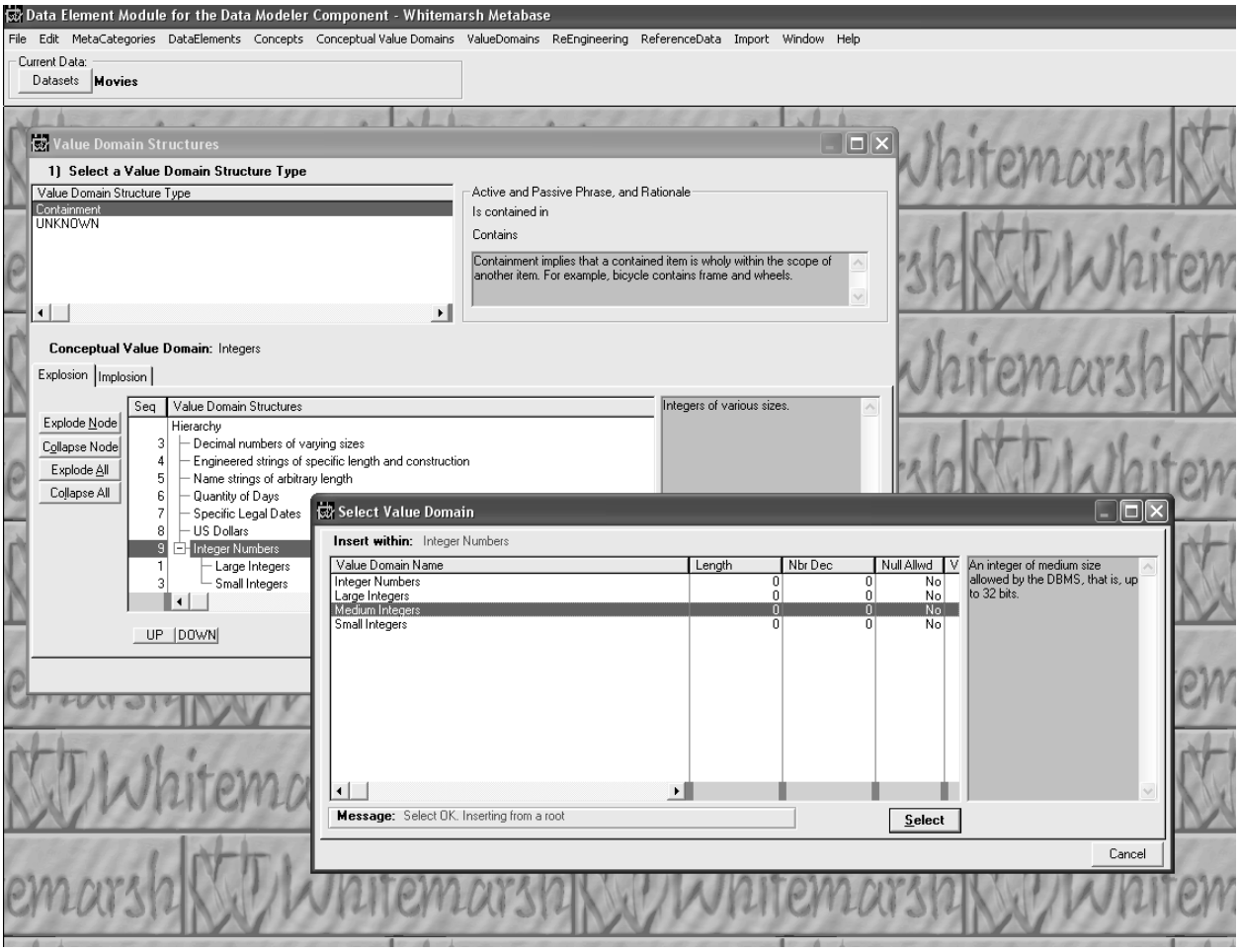

Figure 69. Value Domain Structure update screen.

6.2.5.3 Value Domain Structure Types

The value domain structure type is a way of distinguishing one value domain structure from another. Figure 70 presents the current list. If two different value domain structure hierarchies are interconnected, it may be that the intersection is distinguished from the others by means of a different value domain structure type.

Figure 71 presents the value domain structure type update form. Not only is the name and description of the value domain structure type provided, so too is the active phrase and the passive phrase. The active phrase is employed by the Whitemarsh metabase system when a down-ward structure is presented. For example, <parent> contains <child 1>, <child 2>, ..., <child n>. The passive phrase enables the reverse phrases to be presented. That is, <child 2> is contained in <parent>.

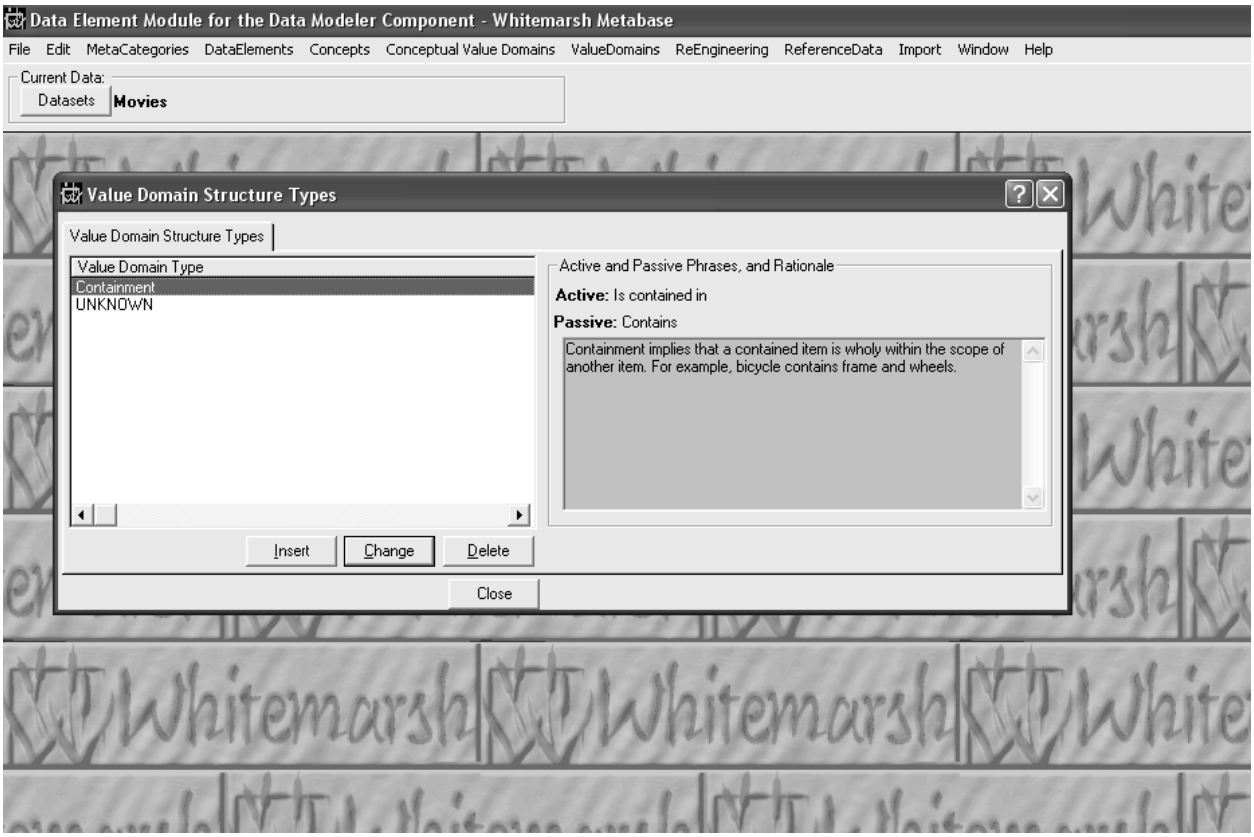

Figure 70. Value Domain Structure Types.

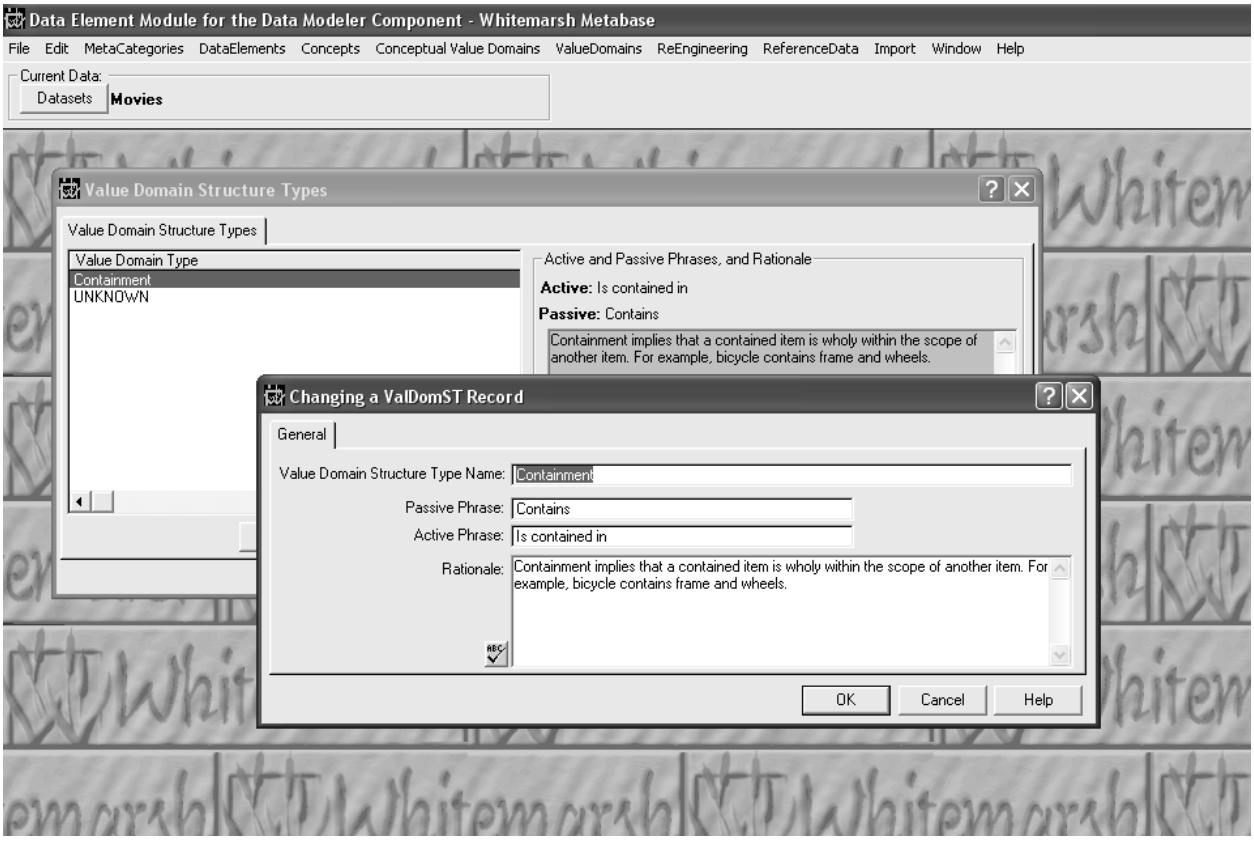

Figure 71. Value Domain Structure Type update screen.

6.2.5.4 Value Domain Data Types

Each value set within a value domain must conform to a particular data type. These data types are analogous to those contained in ANSI Standard SQL. Figure 72 presents the list of value domain value data types and Figure 73 presents the update form for a value domain value data type.

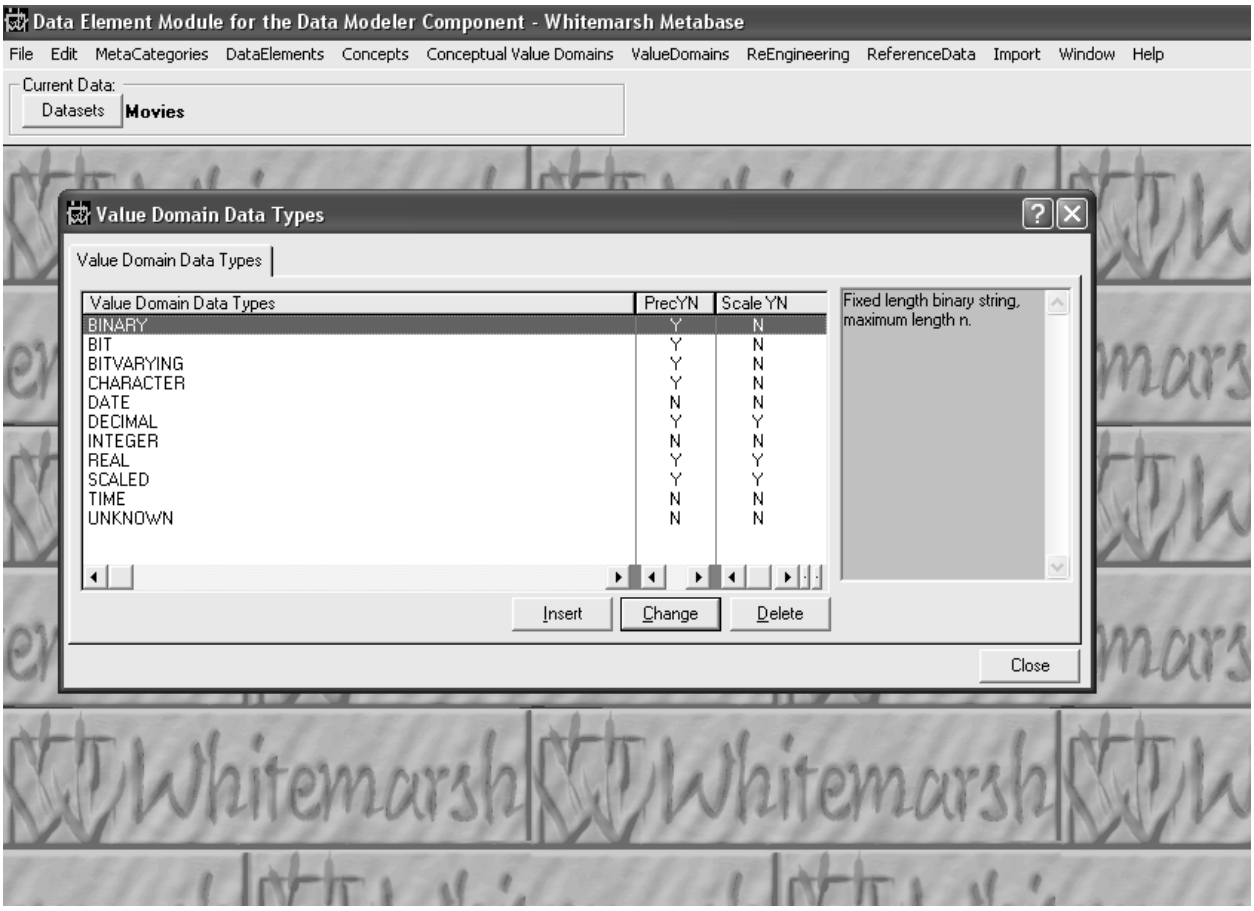

Figure 72. Value Domain Data Types.

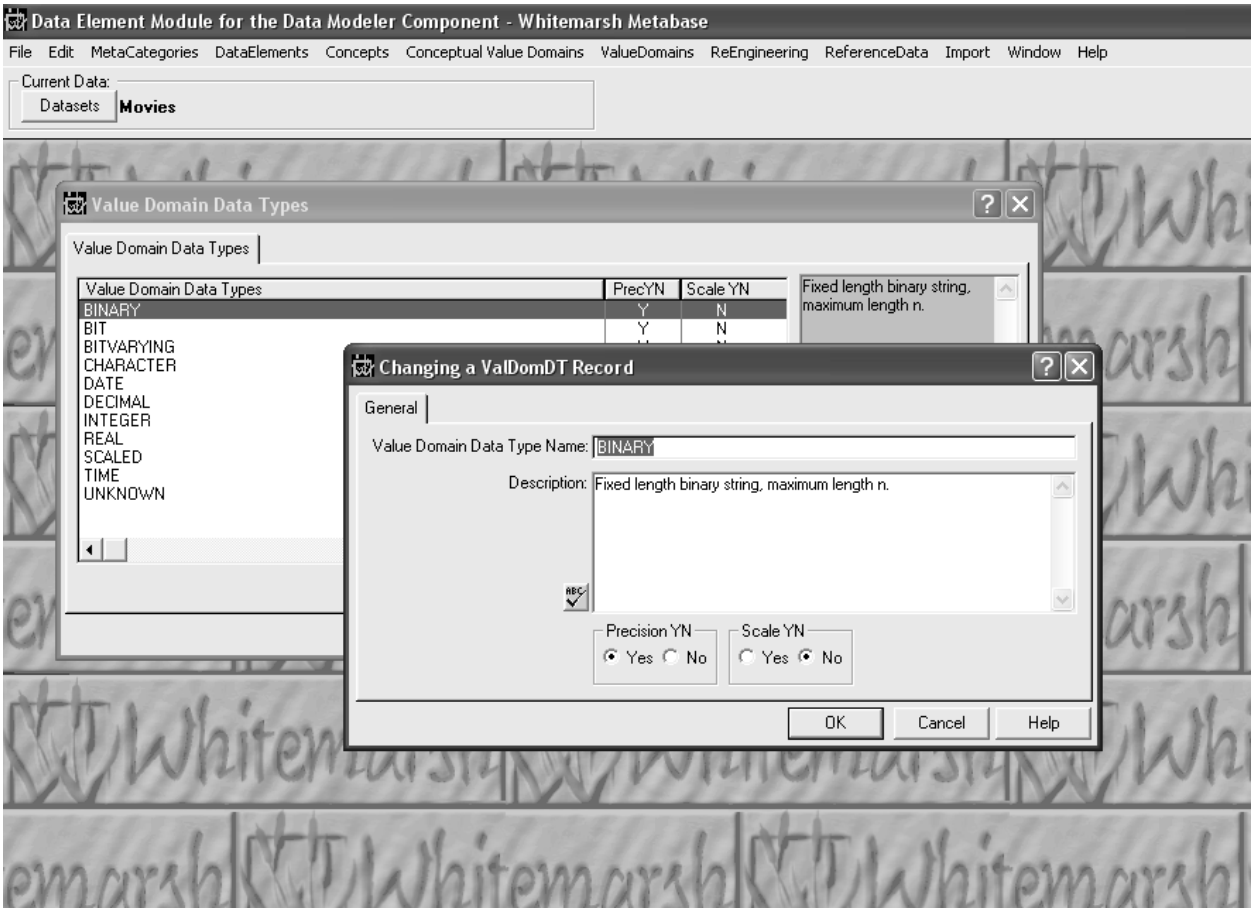

Figure 73. Value Domain Data Type update screen.

6.2.5.5 Value Domain Values

Value domains are not the values. Value domains are the instance of an existence of a set of values. Management of value domain value includes:

- Value domain values
- Value domain structures, and
- Value domain structure types

6.2.5.5.1 Value Domain Values

For any value domain there may be discrete values or ranges of values. Additionally, each set of values may either be the included set or the excluded set. Figure 74 presents a list of values for a highlighted value domain that exists within a structure of value domains. Figure 75 presents the update form for a value domain value.

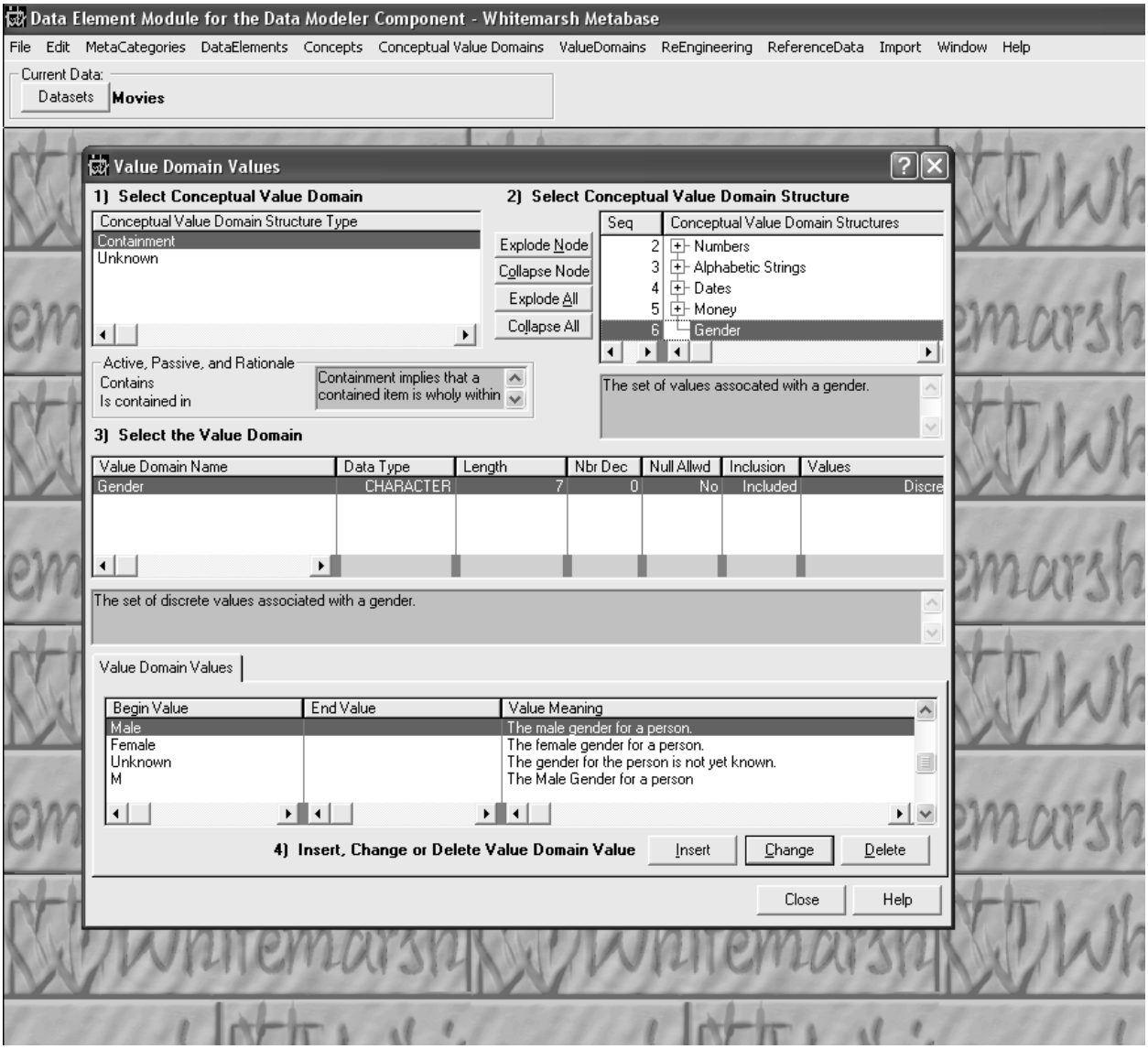

Figure 74. Value Domain Values.

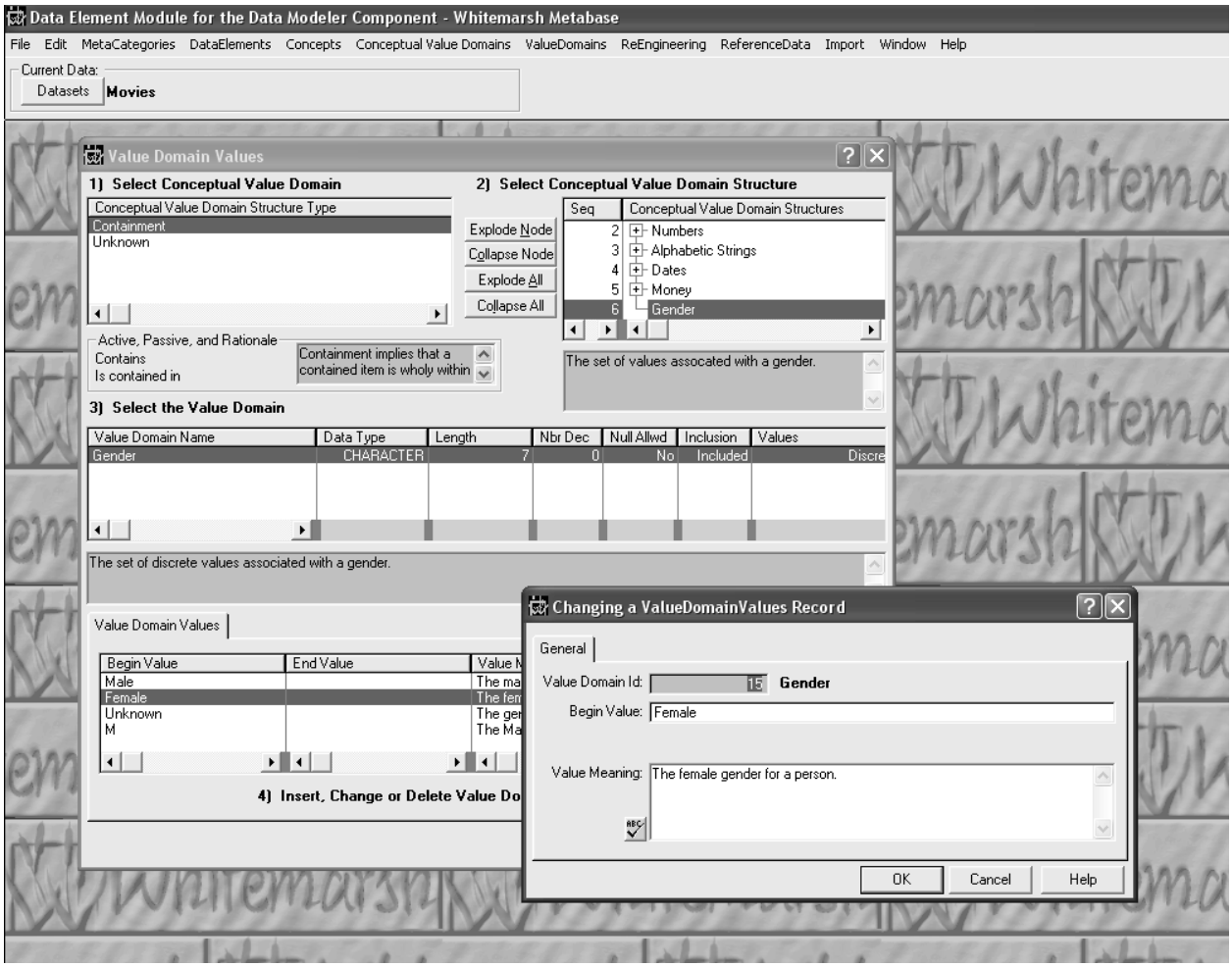

Figure 75. Value Domain Value update screen.

6.2.5.5.2 Value Domain Value Structures

Value domain value structures is a way of expressing relationships among values in different value domains. Suppose there was a general value domain, Genders, and then there were three different collections of value domains, full name (male, female), abbreviations (M, F), or numeric (1, 2). In these instances there would be a need to map the Male => M => 1, and Female =>F=>2. The purpose then of value domain value structures is to support that mapping.

Figure 76 shows the values for the female gender and the relationship between them, that is, Female <is the same as> F. These three are in turn under the name, Gender Value Associations. To add an additional association among Female Value Associations, highlight the appropriate value domain value structure type name and press the "Add new root structure" button. If a leaf value domain value structure type name is highlighted then the button works. Otherwise a message is displayed and the buttons are disabled.

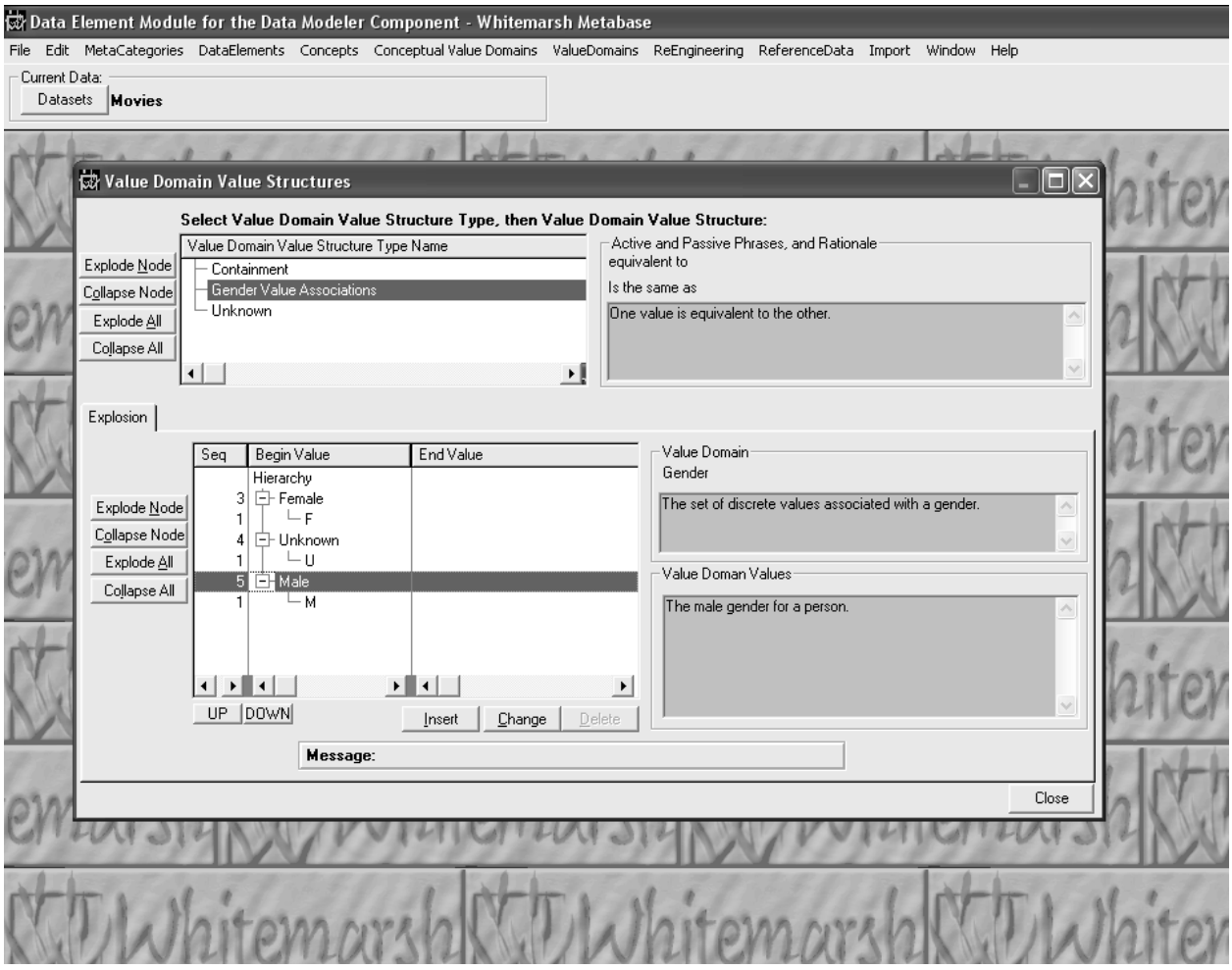

Figure 76. Value Domain Value Structures.

If there is need to map contained values within a value domain then the Insert button can be pressed. That causes the newly inserted value to be shown within the root value. Under either scenario, Figure 77 is presented. This the appropriate value domain value is selected and the select button is pressed. The select button is disabled if the value domain value that is to be inserted causes a twins, recursion or an infinite recursion.

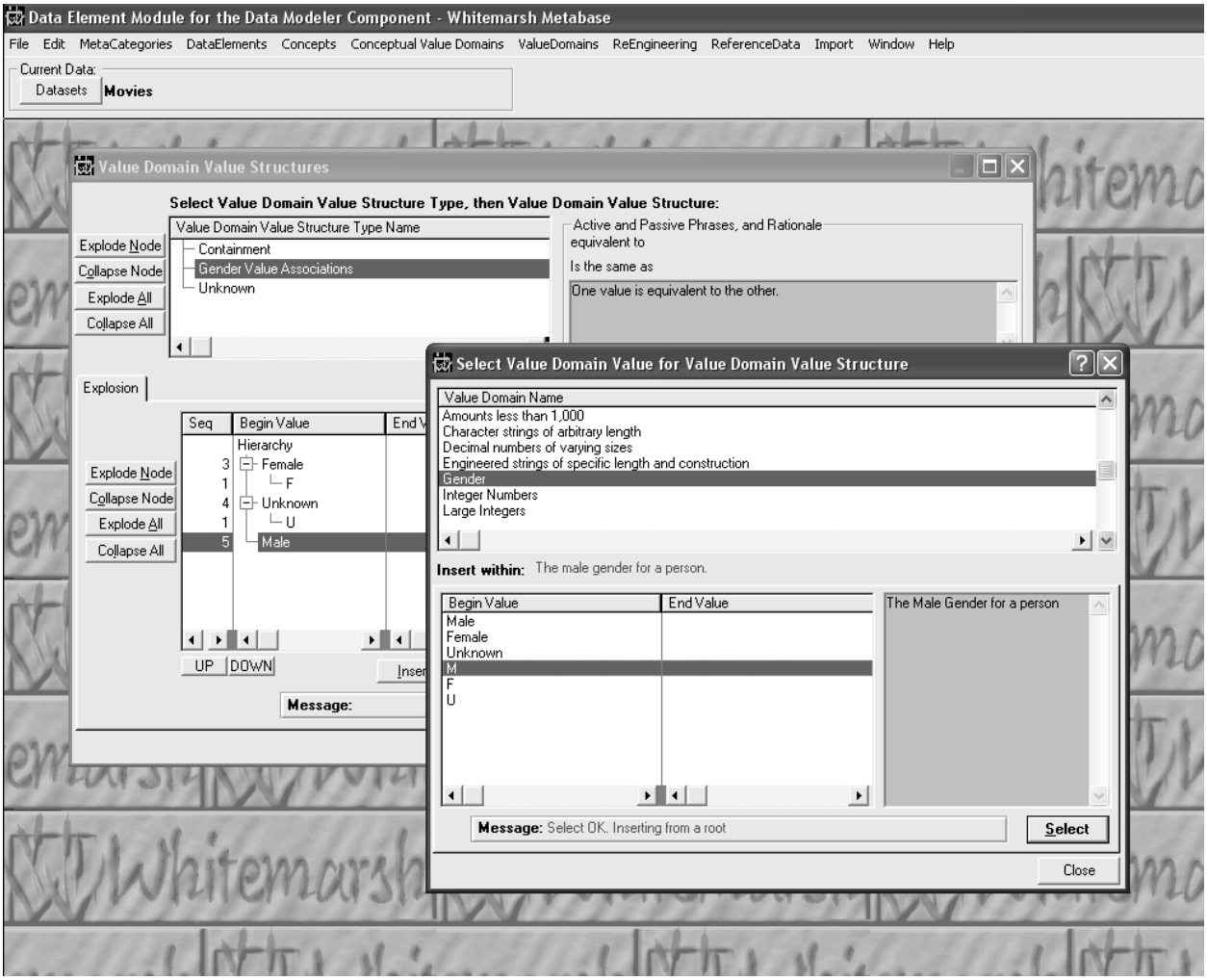

Figure 77. Value Domain Value Structure update screen.

6.2.5.5.3 Value Domain Value Structure Types

The value domain value structure type provides the context for relating the structure of value domain values. Figure 78 shows a hierarchy of Gender value associations and within that male value associations and female value associations. In this example, to insert an entirely new structure, highlight “hierarchy” and press insert. Otherwise, highlight the entry and press insert to make a subordinate entry. In either case, Figure 79 is presented.

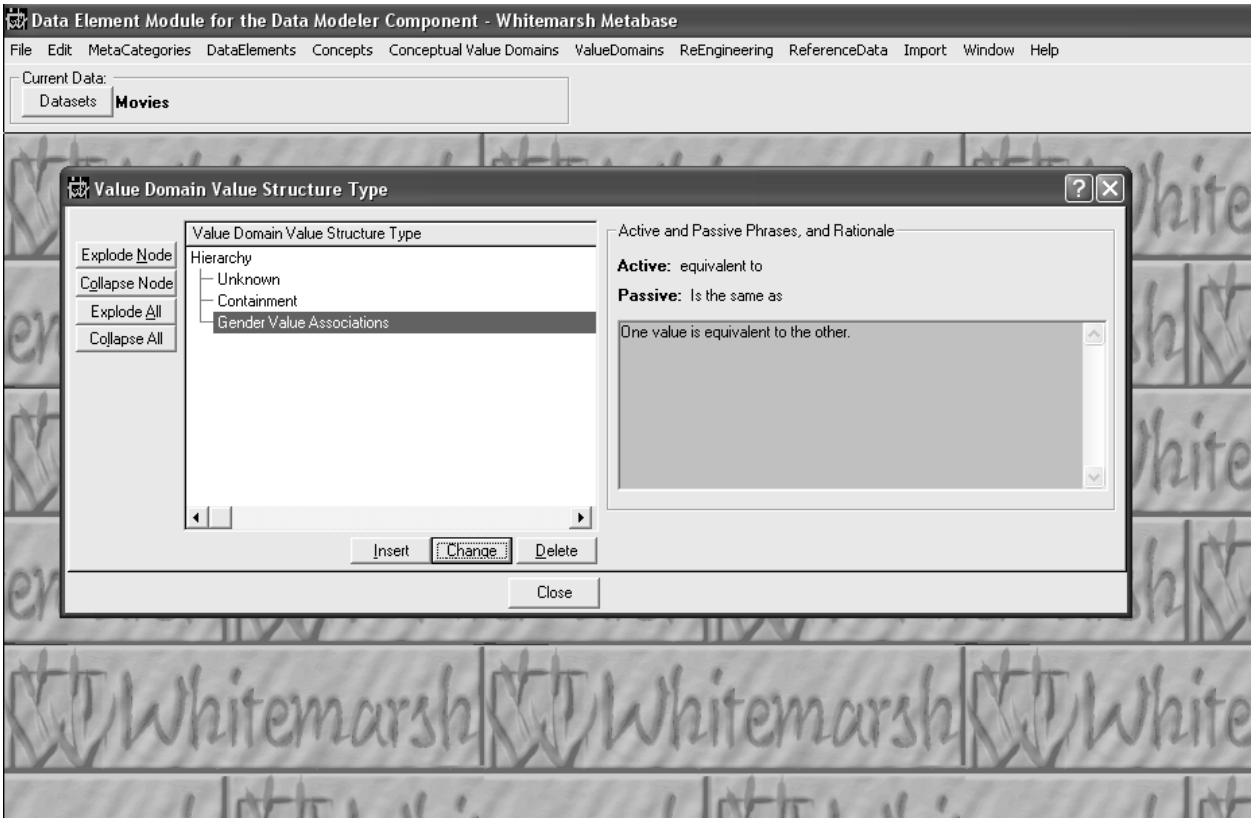

Figure 78. Value Domain Structure Types.

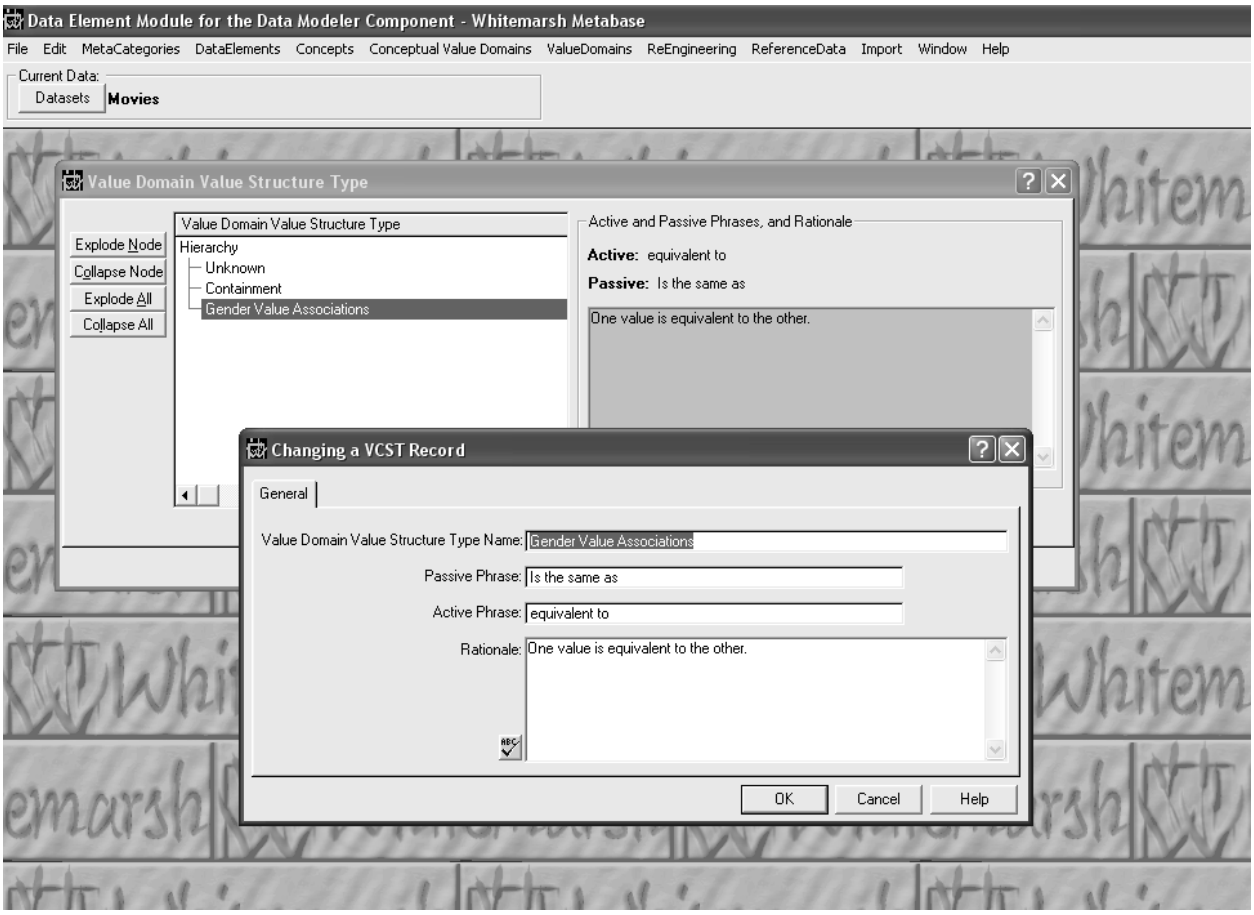

Figure 79. Value Domain Value Structure Type update screen.

6.2.6 Exporting and Importing

The data element module supports exporting and importing data for a number of meta entities. Supported are:

- Concepts
- Data Element Concepts
- Data Elements
- Data Element Classifications
- Value Domains
- Value Domain Values

The rationale for supporting export and import for these is simple: there is likely to already be a list of metadata for each. Since the strategy for exporting and import for each is the same, an example is provided only for data element exporting and importing.

Concepts and Data Element Classifications have no “natural” parents. Thus, their importing is straight forward. Data Element Concepts, Data Elements, and Value Domains all have “natural” parents. When these are imported, their “natural” parents are set to “unknown.” For example, the natural parents for a data element are business domain and data element concept. Upon loading a data element, it’s natural parents will then be the business domain “unknown,” and the data element concept “unknown.” To set these properly, the re-engineering process should be employed to tag the newly loaded data element and its real business domain or data element concept parent.

The table that follows provides guidance as to the fields that should have values for any bulk data loading.

Table for Importing	**Import Columns**	
	Values from Import File	**Valued to “Unknown”**
Concepts	Concept Common Business Name Concept Name Concept Medium Name Concept Short Name Concept Extra Short Name Concept Description	None
Conceptual Value Domain	Conceptual Value Domain Common Business Name Conceptual Value Domain Name Conceptual Value Domain Medium Name Conceptual Value Domain Short Name Conceptual Value Domain Extra Short Name Conceptual Value Domain Description	None

Table for Importing	Import Columns	
	Values from Import File	**Valued to "Unknown"**
Data Element	Data Element Concept Common Business Name Data Element Concept Name Data Element Concept Medium Name Data Element Concept Short Name Data Element Concept Extra Short Name Data Element Concept Description	Concept Structure Id Conceptual Value Domain Structure Id
Data Element	Data Element Common Business Name Data Element Name Data Element Medium Name Data Element Short Name Data Element Extra Short Name Data Element Description	Value Domain Structure Id Data Element Concept Id Business Domain Id
Data Element Classifications	Data Element Classification Common Business Name Data Element Classification Name Data Element Classification Medium Name Data Element Classification Short Name Data Element Classification Extra Short Name Data Element Classification Description	None
Value Domains	Value Domains Common Business Name Value Domains Name Value Domains Medium Name Value Domains Short Name Value Domains Extra Short Name Value Domains Description	Value Domain Data Type ID Conceptual Value Domain Structure Id Value Domain Inclusion is set to 'Included' Value Domain Values is set to 'Discrete'
Value Domain Values	Value Domain Values Common Business Name Value Domain Values Name Value Domain Values Medium Name Value Domain Values Short Name Value Domain Values Extra Short Name Value Domain Values Description	Value Domain Id

The table that follows is a set of five test data records that illustrate the format of data required to load data elements.

'DE CBN Name 1' , 'DE Test Name 1' , 'Medium', 'Shrt' , 'XS', 'Description for DE Test Name 1'
'DE CBN Name 2' , 'DE Test Name 2' , 'Medium', 'Shrt' , 'XS', 'Description for DE Test Name 2'
'DE CBN Name 3' , 'DE Test Name 3' , 'Medium', 'Shrt' , 'XS', 'Description for DE Test Name 3'
'DE CBN Name 4' , 'DE Test Name 4' , 'Medium', 'Shrt' , 'XS', 'Description for DE Test Name 4'

'DE CBN Name 5' , 'DE Test Name 5' , 'Medium', 'Shrt' , 'XS', 'Description for DE Test Name 5'

6.2.6.1 Data Element Exporting and Importing

Data elements may be either exported or imported in bulk. The screens associated with this menu item are clear. The method is through Comma Delimited ASCII. Data elements exist within the following tables:

- Data Element Concept
- Value Domain
- Business Domain

When new data elements are loaded via this facility, the foreign key values for these tables is set to 1, that is, Unknown. Once the data elements are loaded then the reverse engineering facilities need to be employed to make the proper assignments.

For flat ASCII bulk loading, Figure 80 presents the import file selection screen. Once the file name is selected via the ellipsis box, the selected file's rows of data can be viewed via the View button. When the OK button is pressed the file is analyzed to determine if it can be loaded. If it can then Figure 81 is presented. This shows the first record of data, which in this case is a single string, "Check Bank Number."

The next stage in the loading consists of three steps. When "Next Stage" is selected the Step 1, that is, Figure 82, appears. This enables the user to skip the first record, specify a date format (there are not dates in the import unless a full comma delimited structure is being loaded), and to identify the quantity of records to be loaded, where zero means all the records.

Step 2, Figure 83, enables the user to perform duplicate checking. A great deal of caution is recommended here. If duplicate checking is activated then what is checked is the data element's name. In the mapping screen, Step 3, that's the existing column, Derived Data Element Name. This is the only column that is checked for duplicates. To therefore ensure that duplicates are checked then there must be a mapping to this column.

Step 3, Figure 84, is the mapping step. This is accomplished by highlighting a source file field (mouse left button) and then "dragging" that field into on of the "Existing Fields." As the dragging process starts a black arrow appears. Point to just inside the top of the "To" mapping column. When you release the mouse select button the name of the "from" column will appear to the right of the "to" column.

The only step left to do is then press the Start button at the bottom of the screen. All the records will be processed. If there is a duplicate, the system will generate an error message. Accept the error. At the end you will be able to view the duplicate records that were not loaded. Figure 85 then shows the loaded set of data elements.

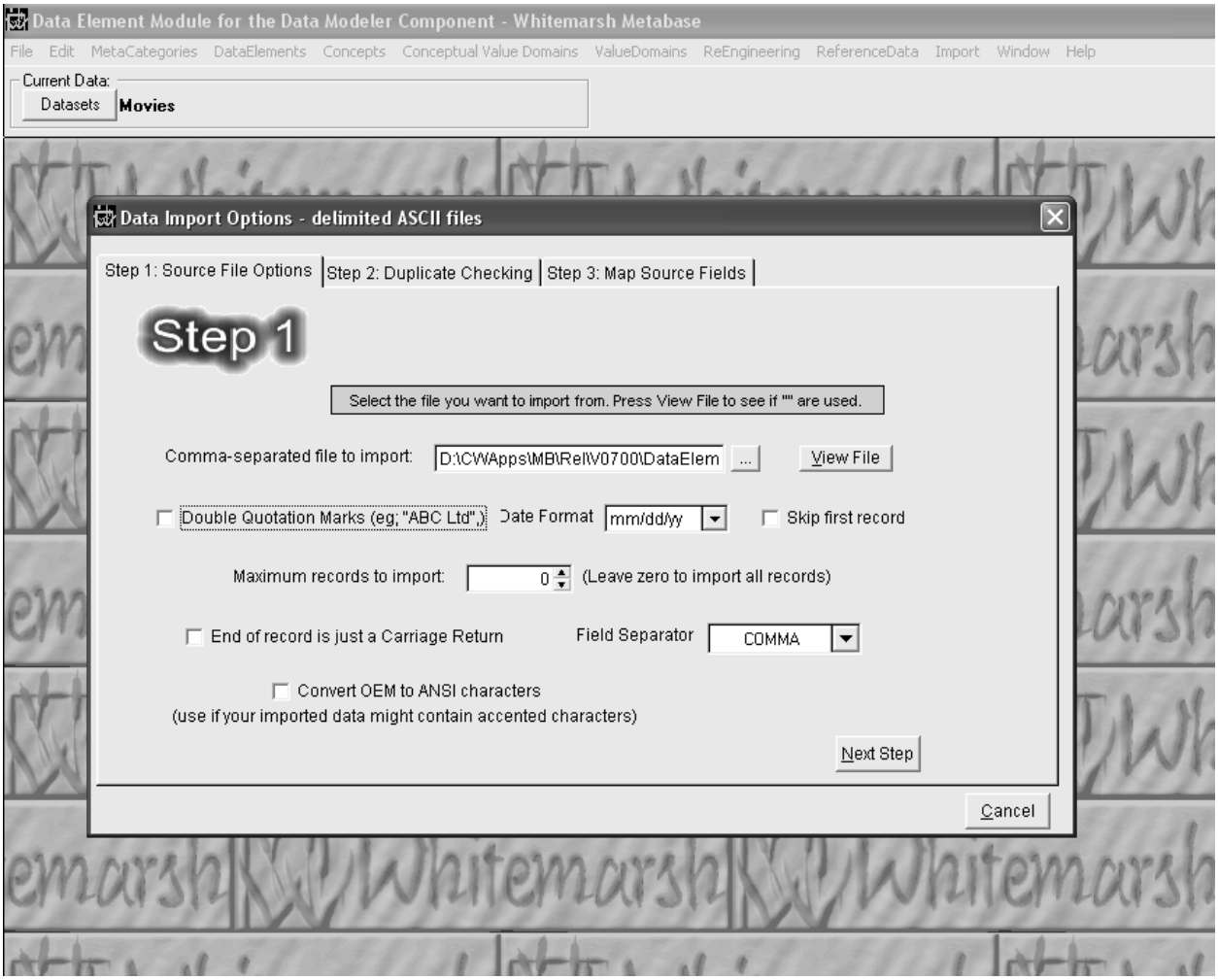

Figure 80. Data Element import file selection screen.

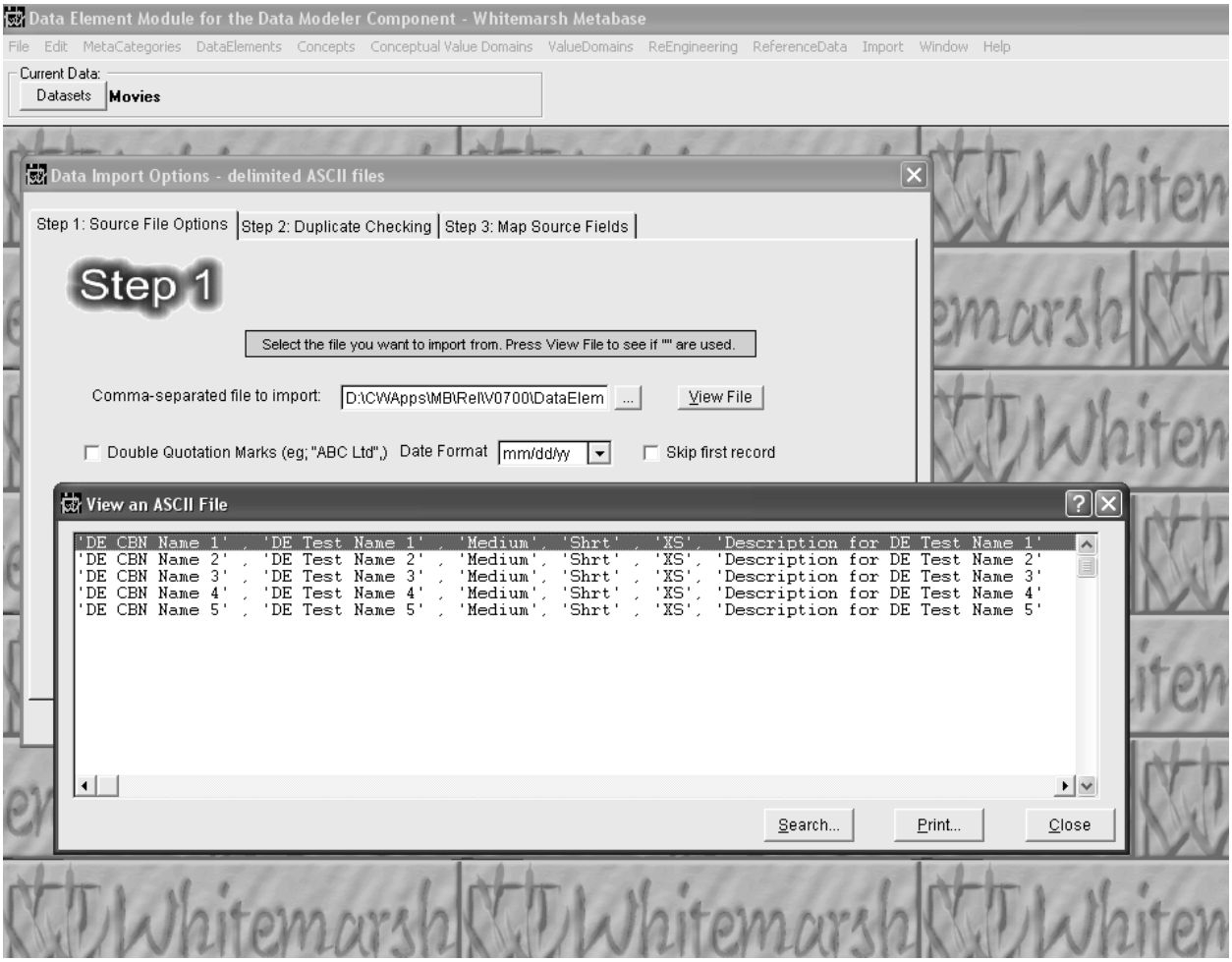

Figure 81. Viewing a selected file for Data Element importing.

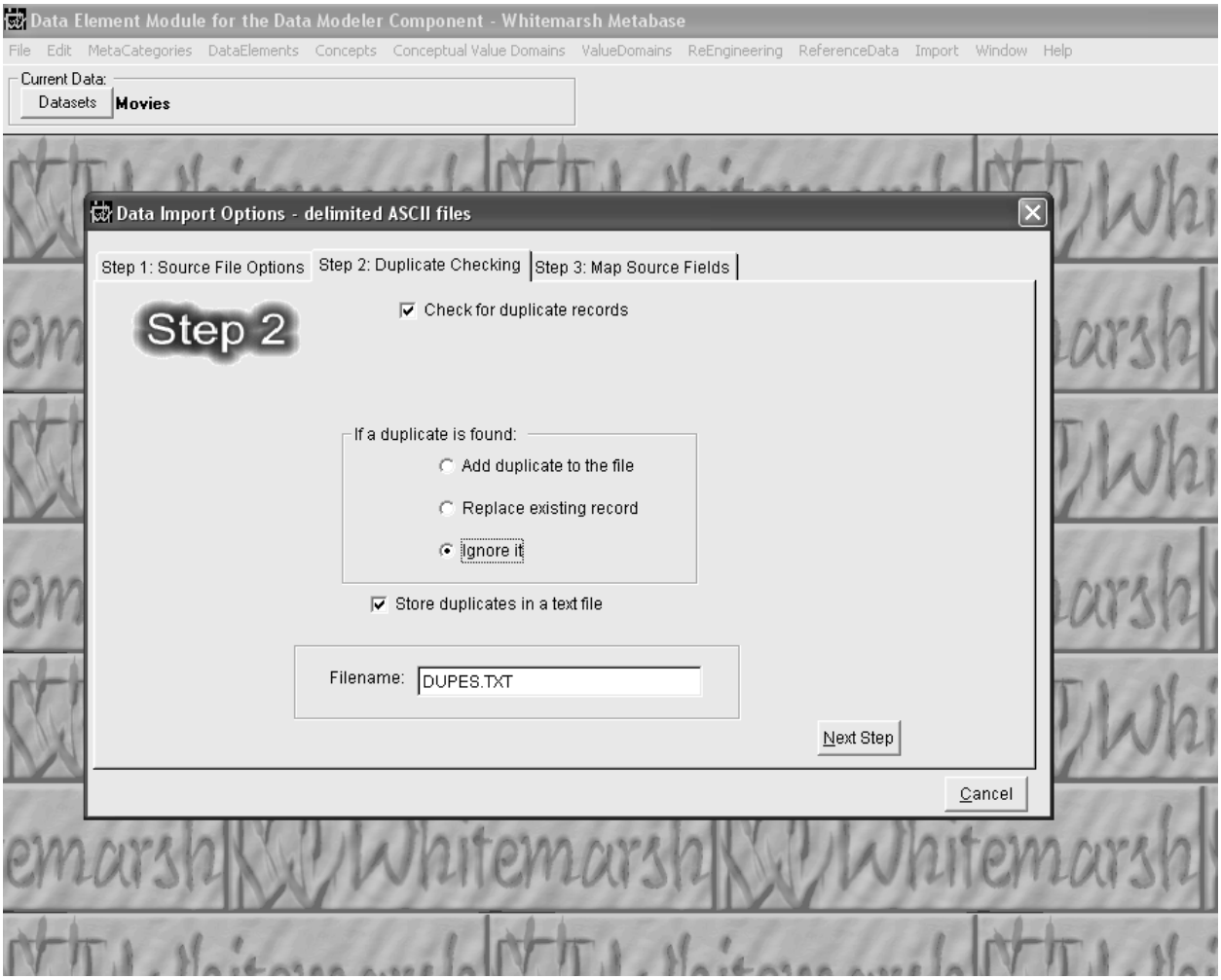

Figure 82. Setting options for duplicate record processing.

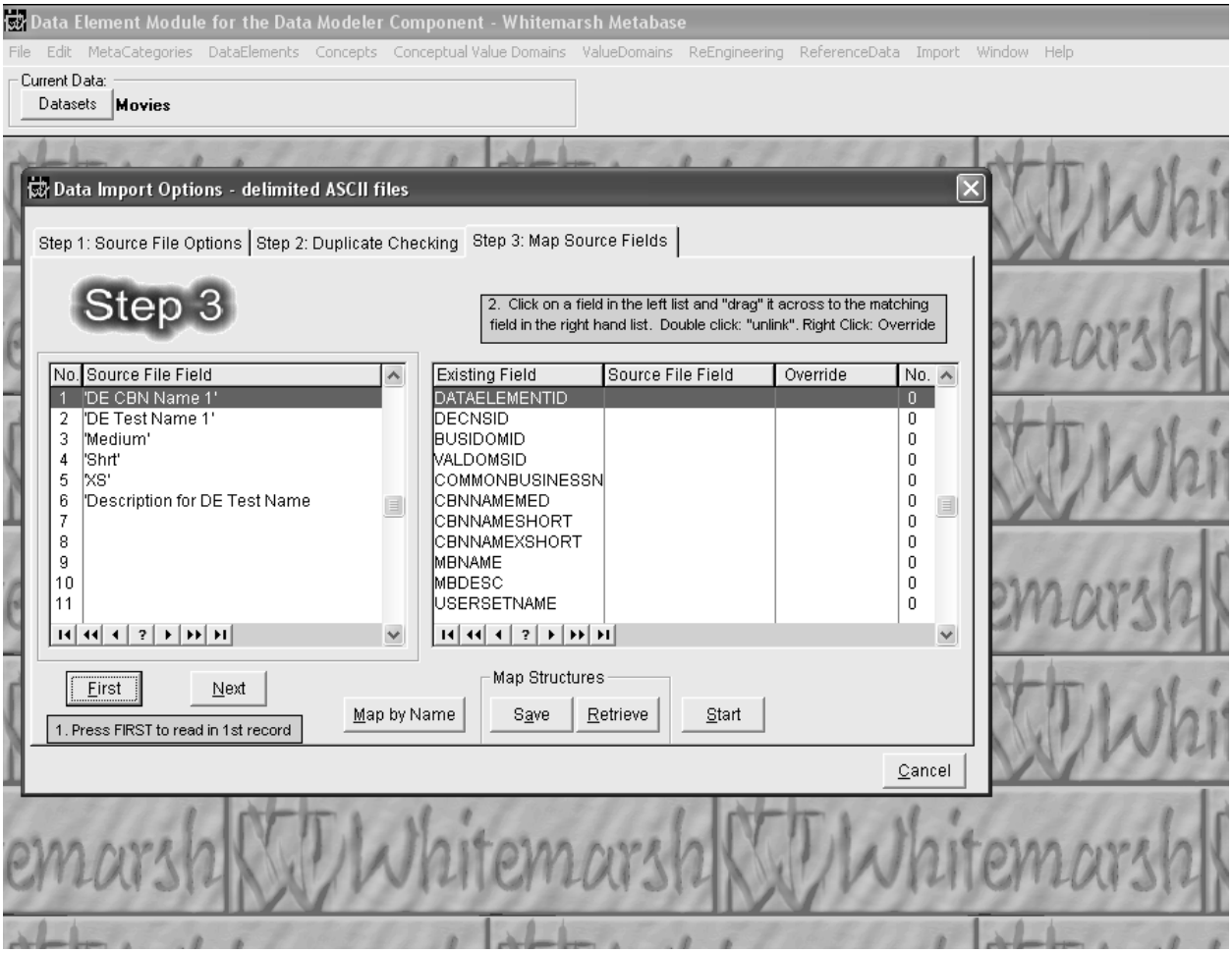

Figure 83. Mapping import file to Data Element columns, initial screen.

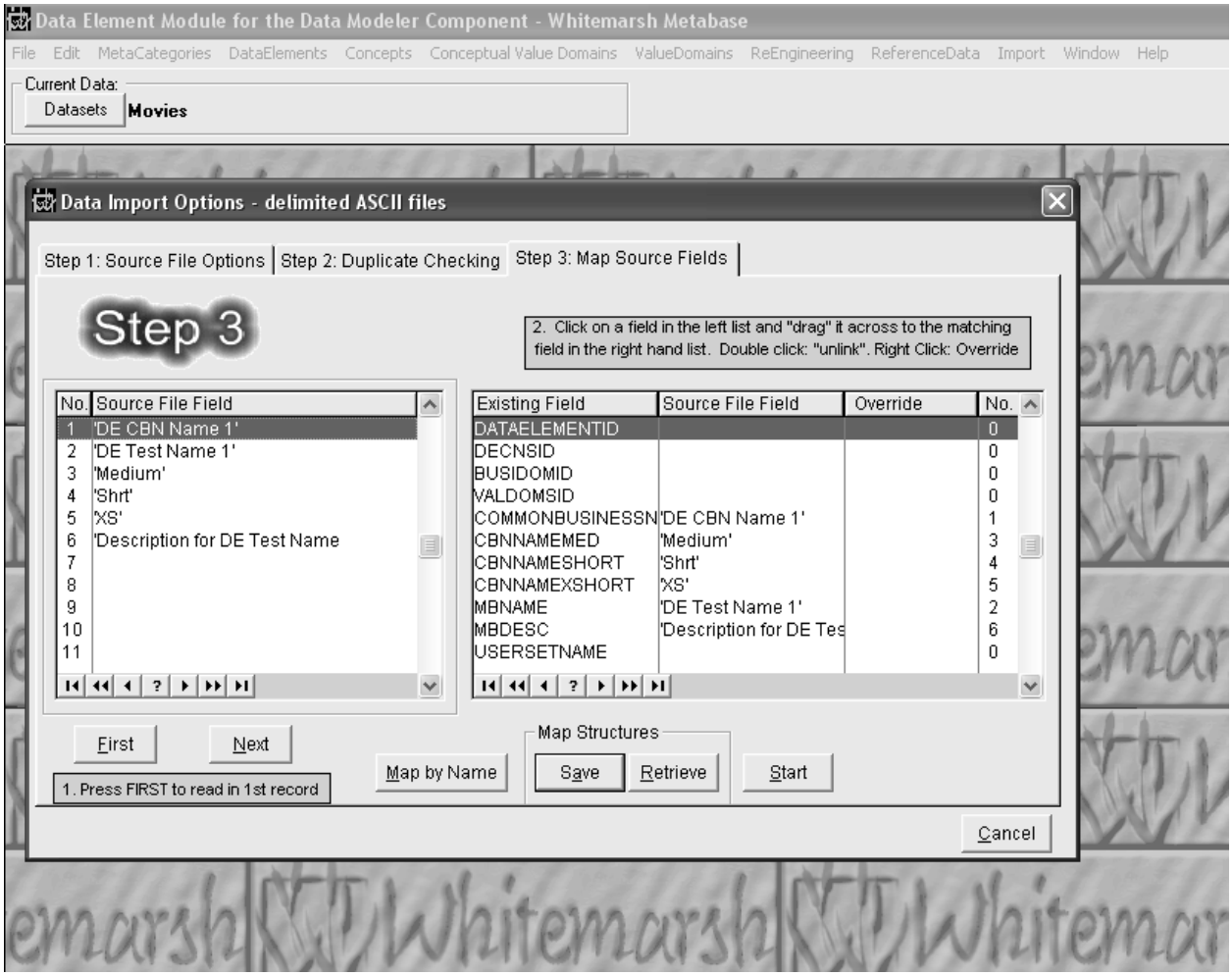

Figure 84. Mapping import file to Data Element columns, final screen.

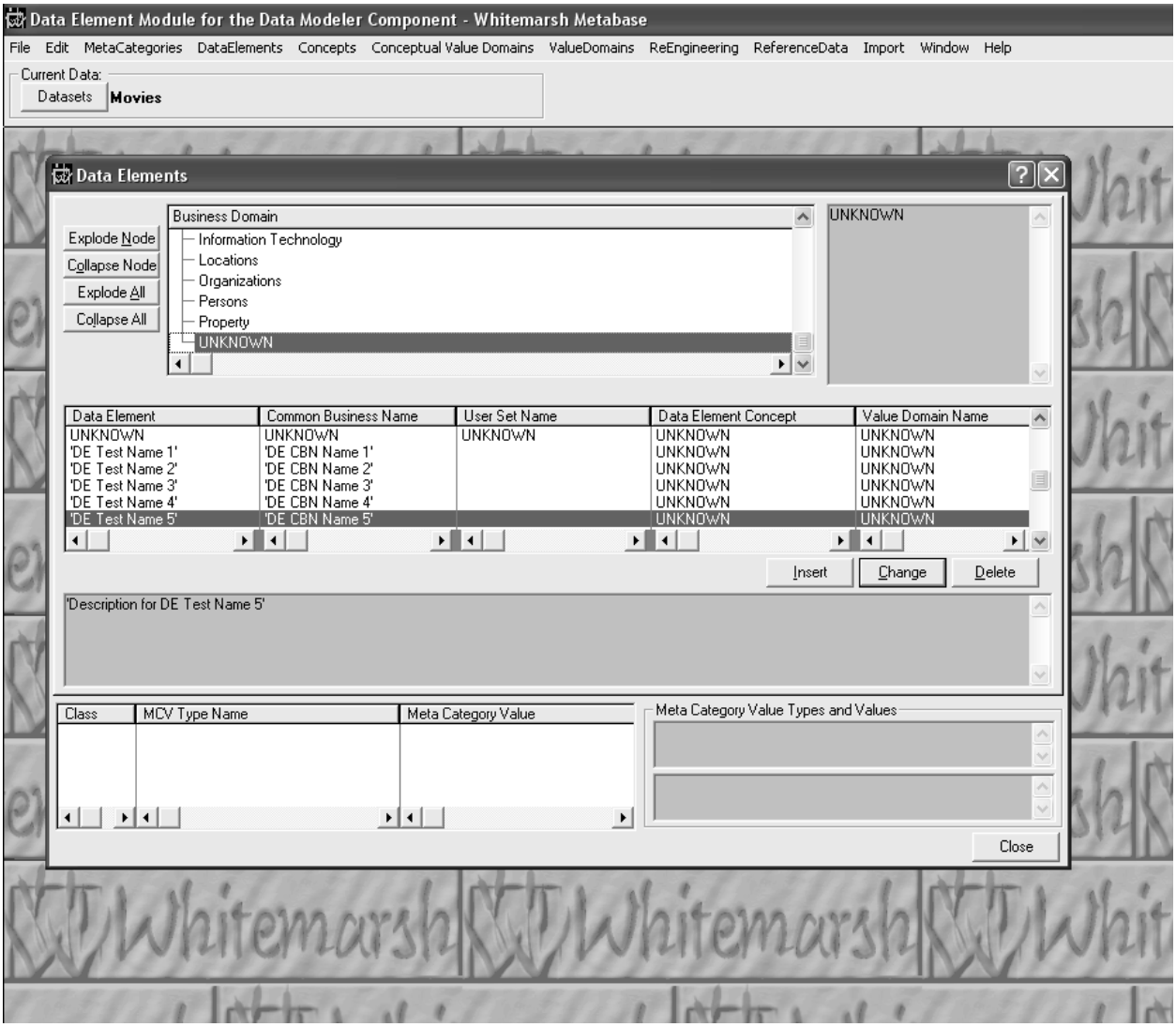

Figure 85. Viewing bulk loaded Data Elements.

6.3 Reports

Reports are accomplished through access to a particular metabase database instance through commercial report writers such as Crystal Reports. Whitemarsh provides about 100 such report templates for Crystal Report access from the Whitemarsh website.