Use the IBM Industry Model Information Insurance Warehouse to define smart and mature data models

Introduction to the data warehousing development method for insurance

Alexander Tarabrin
Hermann Voellinger

December 23, 2010

In this tutorial, understand the method for developing data models for data warehouse projects using the IBM Industry Model Insurance Information Warehouse (IIW), which is part of the IBM Industry Models product defined for the domain of insurance. The tutorial shows the best approach to develop core data warehouse (CDW) models and data mart (DM) models. The tutorial also introduces the recommended data warehousing development method (DWDM) to deal with the IIW model pattern framework to architect DWH solutions for insurance companies.

Before you start

Introduction

Data warehousing design and data modeling is a well-known, significant blend of computer science and IT. The technology grew up in the early 1990s using several approaches developed during that time. The most significant methods were defined by Ralph Kimball (top-down) and W. H. Inmon (bottom-up) (see Related topics).

Commercial data modeling products are valuable because of their content-specific knowledge, which is based on practical experience and business expertise. IBM offers an intellectual capital product family in this space called IBM Industry Models. The IBM Industry Models products consist of mature and well-tested pattern frameworks for data modeling (relational and multi-dimensional), packaged for several industries. This article presents an overview of the Information Insurance Warehouse (IIW), which is a part of the IBM Industry Models product defined for the insurance industry.

This tutorial introduces the method to develop data models for data warehouse (DWH) using the IBM Industry Model IIW. The tutorial illustrates the approach for the development of the core data warehouse (CDW) models (highly normalized data models that hold the atomic data elements) and the data mart (DM) models (de-normalized data models that implement the structure of multi-dimensional data models). Multi-dimensional data models are characterized by the definition of
measures, which are stored in fact tables, and by the definition of dimension tables, which defines the axes or dimensions of the analysis.

The method described in this tutorial is the IIW roadmap for developing data models. The IIW roadmap is based on the top-down approach, which starts with the capture of business requirements and the definition of the business model (in terms of the IIW known as the analysis data model). Defining the business requirements is the prerequisite for all further work. Ideally, this work should be carried out as a joint effort of the data modeler and experts from the business departments. When the business departments create and approve the model, the phase to create logical models starts.

The design of logical models consists of two steps: designing the DWH logical model (CDW) followed by designing the DM logical model. It is important to follow this sequential approach. Starting succeeding phases before finishing predecessor phases might yield undesirable results. Therefore the IIW roadmap's structure, and this tutorial, is divided into the following four phases:

1. **Phase 1: Capturing IIW business requirements**
2. **Phase 2: Defining the analysis data model**
3. **Phase 3: Designing the data warehouse logical model**
4. **Phase 4: Designing the data mart**

These four phases fulfill different goals and offer different deliverables:

**Phase 1: Capturing IIW business requirements**
- A complete description of the business requirements that the BI project should solve. The deliverables are a conceptual model and an analytical requirements model.
  - **Conceptual model**
    - A model of all the concepts and business terms to be used across the organization
  - **Analytical requirements model**
    - Predefined models of business requirements that address specific industry issues. Models are expressed as measures and dimensions

**Phase 2: Defining the analysis data model**
- A conceptual model that represents an ideal picture of the business concepts and how these concepts relate to each other. This model is platform independent and does not require physical aspects of the implementation. The deliverable is the analysis data model.
  - **Analysis data model**
    - A data model that specifies the normalized data structures required to represent the concepts defined in the conceptual model.

**DWH and DM design phases**
- The business concepts mapped on an entity-relationship (ER) logical model (DWH) and on a multi-dimensional (MD) logical model. These models are the basis for the physical structure of the data in the database. The deliverables are DW design data models and DM design data models.
  - **DW design data models**
    - Data models that represent the enterprise-wide repository of atomic and analytical data used for informational processing
DM design data models
Dimensional models that implement analytical requirements and are structured to allow specific dimensional analyses
Figure 1 summarizes these deliverables.

Figure 1. Deliverables of the four IIW phases

IIW also defines three model layers:

- The foundation layer contains the conceptual and analytical requirements models.
- The analysis layer covers the analysis data model.
- The design layer contains the DW design and the DM design models.

The diagram in Figure 2 depicts these layers.

Figure 2. IIW model layers

The following sections of the tutorial describe the four phases with examples of each phase using InfoSphere™ Data Architect (IDA). The examples use IBM IIW Model Version 8.2. The IIW model
content is imported in IDA with the help of the Enterprise Model Extender (EME) tool. EME is a set of plug-in extensions to the IBM InfoSphere Data Architect product. To follow along with the tutorial, you will need these products installed.

Phase 1: Capturing IIW business requirements

The need for appropriate tooling and model patterns appears in the early phase of requirements analysis. The work packages of this phase are assigned to business analysts with deep industry knowledge and high-level IT skills. The deliverables of such work packages are often non-structured documents, such as meetings protocols, presentations, and word processor documents. The information collected in this way often does not have any independent models. The information is delivered as is for the next phase (the business model), which creates the overhead of manual processing of large numbers of input deliverables. There is a clear need to introduce high-level, non-entity-relationship models.

Access to the knowledge and reference solutions is another requirement. The available solution patterns should be represented in an appropriate way to allow business consultants an efficient and easy way to do the following:

- Access the IIW framework patterns and documentation
- Collect requirements in the IIW framework
- Map deliverables from the requirements analysis phase to the components of the IIW framework

The IIW introduces the foundation layer models to provide business consultants and analysts with the toolset and intellectual property resources, which are distributed as commercial product. The foundation layer models include the following:

**Conceptual model**
- Contains the business metadata to be used in other models

**Analytical requirements model**
- Uses the business metadata identified in the conceptual model to document the requirements of analytical queries

**Business concepts vocabulary**
- Delivers the dictionary-like representation of collected business terms

**Conceptual model**

The conceptual model is the highest model for the definition of business requirements. The business data collected in this phase are fairly unstructured. The conceptual model delivers the tool support to collect the information about numeric and non-numeric data elements and dependencies between them.

The parts of the conceptual model are:

- Aggregate descriptors that focus on measure-based requirements. The aggregate descriptors are references as measuring templates for the analytical requirements model.
• The concepts part of the model, which is focused mainly on the collection of hierarchically ordered metadata. This part is used in the analytical requirements model for the definition of dimensions.

The aggregate descriptors model represents a list of pre-defined descriptors. Each descriptor contains a unique label and documentation text, as shown in Figure 3.

**Figure 3. Properties of the Accounts Receivable aggregate descriptor**

(View a [larger version](#) of Figure 3.)

For calculated measures, you can use a feature to document the calculation expression and reference related measures. An aggregate descriptor can be linked to another model (for example, the analytical requirements model or analysis data model).

The concepts part of the model contains the following types of metadata:

**Descriptors**
- Can be numeric or non-numeric. The numeric descriptors can be used for calculated measures.

**Relationships**
- Describes relationships between concepts.

**Hierarchies**
- Describes the order of business concepts. Each concept has several sub-concepts that classify and describe the concept. The following three sub-concepts are used:
  • Classifier
  • Descriptor
  • Relationship

By using classifiers, you can document the business hierarchy. For example, an account has four classifiers:
  • Accounting period type
• Account status reason type
• Account status type
• Account type

Under account type, you can define four subtypes, including monetary account. Monetary account itself has seven subtypes, and so on, as shown in Figure 4. This art of diagram notation is called schema-value notation. Using this notation, you can document the metadata (level names) and business data (level values) in the same hierarchy.

**Figure 4. A cut-out from the conceptual model depicting a fragment of the account hierarchy**

The conceptual model is an important feature of the IIW for documenting business concepts and their dependencies at a high business-level view. It also shows the existing dependencies and relationships of a business object, such as the business object account in Figure 4.

**Analytical requirements model**

The purpose of the analytical requirements model (ARM) is to group and classify the concepts collected in the conceptual model. An analytical requirements model represents single analytical requirements. An analytical requirement contains measures and dimensions needed to analyze a particular business case.

Analytical requirements are grouped into several groups called focus areas. Each focus area represents a domain of business problems, such as claims analysis, product management, risk management, and so on. You can create substitute focus areas. For example, claim efficiency analysis can be a substitute for claims analysis, as shown in Figure 5.
A focus area contains analytical requirements. An analytical requirement contains a set of measures and a set of dimensions. For example, the requirement *Customer risk analysis* contains the dimensions *Policy*, *Product*, and *Time dimension*, and it contains the measures *Number of accidents* and *Number of claims*, as shown in Figure 6.

The links between the analytical requirements model and the conceptual model are crucial. All measures and dimensions are linked to appropriate concepts in the conceptual model. Measures reuse aggregate or measurable properties that are pre-defined in the conceptual model. Dimensions are views or wrappers of business concepts, classifiers, relationships, or descriptive properties that are pre-defined in the conceptual model.

The advantage of this linking approach is that semantically identical measures or dimensions referenced in different analytical requirements are mapped to the single concept in the conceptual model. This solves the problem of redundancy and enables putting the documentation about a particular business concept in a single place. Conversely, you can inspect any element of the conceptual model for references from the analytical requirements model.

IIW is delivered with a set of pre-defined analytical requirements grouped into focus areas. Business users can select the analytical requirements to modify or create new analytical requirements. See the article *Scoping the IBM Industry Model for banking using Enterprise Model Extender and InfoSphere Data Architect* for more information.

**Business vocabulary**

The business vocabulary (or business glossary) is a required deliverable of any data warehousing project. Often it is maintained manually as a semi-structured document with dependencies on the requirements catalog and data models. The IIW delivers the business vocabulary as an integrated part of the modeling environment.
The business vocabulary summarizes the business terms collected in the conceptual and analytical requirements models. The business vocabulary consists of defined terms (words) grouped to dictionaries, as shown in Figure 7.

**Figure 7. Business vocabulary**

![Business vocabulary diagram](https://via.placeholder.com/150)

(View a larger version of Figure 7.)

Each word is provided with the description, which is the description of an appropriate element of the conceptual or analytical requirements model. The business vocabulary is open for modifications. Business users can edit or create dictionaries and words. Business users can define a word life cycle in which users can specify status, such as candidate, accepted, standard, and deprecated.

The business vocabulary offers several other features, including the capability to specify synonyms or related words.

You can export the content of the business vocabulary to IBM InfoSphere Business Glossary using the Metadata Server.

**Summary of phase 1**

This section gave an overview of the major models of the IIW foundation layer. The models are the framework for collecting the data during the requirements analysis. The models are the deliverables for the next phases.

**Phase 2: Defining the analysis data model**

IIW uses the term *analysis data model* to refer to the *business model*. The data warehousing development method (DWDM) delivery approach foresees the creation of the analysis data model before the creation of the logical and physical models. The analysis data model is business oriented. It is independent of any design or architecture. Defining the analysis data model is one of the most resource-intensive phases of the design process.

The analysis data model is a data model that specifies the normalized data structures required to represent the concepts defined in the conceptual model. As an analysis model, the analysis data model does not add any new business concepts to the content defined in the conceptual model.
Analysis business model structure

The analysis data model consists of 23 business areas, including party, place, claim, event, and so on. The analysis data model has over 1100 entities. There is a type hierarchy for each core entity, such as the type hierarchy for Claim core entity, as shown in Figure 8.

Figure 8. Claim type hierarchy

The supertype and its subtypes (known as natures) are connected using the parent-child relationship. The attributes are held in the appropriate level of the hierarchy.

The entities and attributes of the analysis data model are described with Enterprise Model Extender (EME)-specific properties. Additional Classification and Mappings tabs are shown in Figure 9.
Figure 9. EME-specific properties

Using the Classification option, you can verify the appropriate type of entities and attributes. The most common types are semantic entity for entities and semantic attribute for attributes. All attributes of the analysis data model are mapped to the conceptual model, as shown in Figure 10.

Figure 10. Mapping to the conceptual model

Model verification and analysis

You can verify and analyze the model to identify the erroneous modeling or gaps. Model analysis is defined as the set of rules that can be checked against the analysis data model you created. For example, there is a rule to check for the existence of mappings for all attributes defined in the model, as shown in Figure 11.
Use the IBM Industry Model Information Insurance Warehouse to define smart and mature data models

Figure 11. Model analysis rules

If some mappings are missing, as shown in Figure 12, the appropriate error message appears after you execute the model analysis.

Figure 12. Error message sample

Summary of phase 2

This section described how to define the analysis data model (business model), which is one of the most important tasks in the DWDM method. The person assigned to this activity needs both business knowledge and data modeling skills. The analysis data model is the basis for the definition of the logical data model described in Phase 3.

Phase 3: Designing the data warehouse logical model

Business users complete the phases described in the first two phases of this tutorial. Designing IIW logical models is rather IT-oriented, and it is better-suited for IT users (data designers) to complete. The design layer of IIW consists of the atomic part (the enterprise model) and of the analytical part (conformed dimensions model and data mart models). This section describes the IIW enterprise model.

The structure of the enterprise model is similar to the analysis data model. The model is packaged into several packages. Each package consists of a set of artifacts (entities, attributes, and relationships) and some diagrams for visualization needs, as shown in Figure 13. This is the standard structure of logical models in IBM InfoSphere Data Architect.
Enterprise model artifacts

The data warehouse model is a data model that represents the enterprise-wide repository of atomic data used for informational processing. This model includes the history of the value changes of business information that may vary over time. You might want to keep track of this history for analytical purposes.

The data warehouse model defines the following attribute types:

**Fundamental entity**
Contains atomic business information. A fundamental entity is either versioned or non-versioned. It requires either an anchor entity if versioned or a root entity if non-versioned to maintain its versions.

**Anchor entity**
Acts as an anchor to maintain different versions of an entity instance. Anchors are also used as time-invariant keys (TIKs) in a data warehouse environment.

**Root entity**
Acts as super-type for non-versioned fundamental entities.

**Associative entity**
Serves as relationship between two root or anchor entities.

**Population characteristic entity**
Contains information regarding ETL jobs.

**Classification entity**
Instantiates specific semantic attributes of data type *enumeration*.
The entities are connected using relationships of different types as follows, defined for the enterprise model:

**Design relationship**
Connects two entities, usually one-to-many, for design purposes (as navigation or better performance).

**Anchor relationship**
Connects the fundamental entity with the related anchor entity.

The entities have attributes. The attribute types defined in the enterprise model are as follows:

**Basic attribute**
Contains business data.

**Candidate key**
Contains the business key that uniquely identifies an entity instance.

**Relationship attribute**
Contains a foreign key attribute.

**Derived attribute**
Contains a value derived (or duplicated) from the value of one or many of the other attributes.

Figure 14 shows a cutout from the data diagram of the claim domain. The attributes are omitted to improve the readability.

**Figure 14. Cutout from the enterprise model for claim domain**

![Cutout from the enterprise model for claim domain](image)

**Customization approach**
The approach of the enterprise model customization is described in detail in the online help installed with Enterprise Model Extender. There are specific techniques to add or modify an entity, relationship, or attribute of any type. The techniques are figured out in the form of eight transformation rules. The rules describe the customization approach by adding semantic association entities and semantic associations; transforming super-semantic entities and sub-semantic entities; and coping with derived and duplicate attributes.

When customizing the enterprise model, the data designer should keep in mind that the elements of the enterprise model should be linked to the appropriate elements of the analysis data model.

**Towards a physical model**
Unlike the analysis data model, the enterprise model is IT-oriented. The enterprise model contains the following features:
Data versioning

One of the most important aspects of the data warehousing design. The fundamental entities of the enterprise model (EM) are featured with the following four attributes:

- EFFECTIVE_FROM
- EFFECTIVE_TO
- VALID_FROM
- VALID_TO

The EFFECTIVE_FROM/TO attributes rely on the period of semantic validity. The VALID_FROM/TO attributes rely on the timestamps when the data have been populated from the source system and represent the period of technical validity.

Primary keys

The primary keys defined in the EM are 4-bit integers. They are surrogate keys without business semantic. They are not derived from the primary keys of the source systems.

Data population

The entities contain the data population attributes that describe the information about the ETL process.

Summary of phase 3

This section described the enterprise model and the approach to customizing it. You can transfer the enterprise model to the physical model using the InfoSphere Data Architect wizard.

Phase 4: Designing the data mart

Business users complete the phases described in the first two phases of this tutorial. Designing a data mart is rather IT-oriented, and it is better-suited for IT users (data designers) to complete. The design layer of IIW consists of the atomic part (the enterprise model) and of the analytical part (conformed dimensions model and data mart models). This section describes the analytical part.

The conformed dimensional model is the enterprise-wide repository for analytical data. It contains dimensional data structures based on fact entities that enable easy distribution of analytical data to downstream analytical models, such as data marts.

The structure and business content of the dimensions in the conformed dimensional model are based on and re-use the data warehouse model. The measures in the fact tables support the measures defined in the analytical requirements.

In addition to the data structures and content originated from the data warehouse model, the conformed dimensional model includes the following:

- Additional relationships to build the aggregation paths on which measures can be analyzed
- Additional entities for efficiency of storage and analysis, such as profile and helper entities

In InfoSphere Data Architect, there are several pre-defined IIW data marts logical models, as shown in Figure 15.
Use the IBM Industry Model Information Insurance Warehouse to define smart and mature data models

Figure 15. Sample cut-out from the IIW data mart structure

Each data-mart model itself can consist of a set of analytical subsets. Figure 5 shows that claim efficiency analysis consists of four analytical packages (hold data type) and one association package (hold relationship entities). This is the standard structure of logical models in IBM InfoSphere Data Architect.

Using the conformed dimensional model

The conformed dimensional model is like a super data mart. However, consider end-user queries with caution because of the following reasons:

- The conformed dimensional model's dimensional structures are organized as snowflakes, which make the queries more complex.
- Some fact entities might be too general, which can result in business measure names that are less meaningful.
- The number of dimensions per fact entity might be too high. The granularity of dimensions is as low as possible to allow for most granular analytical requirements without requiring any design maintenance. This could potentially result in too many facts and bad response times on queries.

The conformed dimensional model is based on the following two main design principles:
Conformed dimensions
A conformed dimension is a master dimension for which all parties in the enterprise have agreed to the contents. Conformed dimensions enable re-usable aggregation paths for measures across multiple fact tables.

Conformed facts
A conformed fact is a measure for which all parties in the enterprise have agreed to the business definition. The fact can be used in analytic computations across separate data sources and with other conformed facts.

The two design principles define consistency across fact tables, improve the quality of analytical results, and facilitate analysis techniques, such as drilling across.

The conformed dimensional model is partly denormalized to ease extracting the dimensional structures to populate downstream data mart models.

Artifacts of the conformed dimensional model

When using IBM Industry Models to develop BI solutions, the conformed dimensional model uses the following artifacts:

Fact entity
An aggregate entity that regroups a set of measures (facts) that all share the same dimensions. The primary key of a fact entity is defined as the concatenation of all foreign keys of the entities that are used as its dimensions. The fact entity is the core entity of a dimensional data structure. It can serve as the basis to create subsets of the structure that are distributed to data marts.

Helper entity
A helper entity manages analysis of a complex hierarchical structure of variable depth of a dimension. A helper entity can be used only if the hierarchical structure is a tree and not a network. This entity contains one instance for each separate path from each node in the hierarchical tree to itself and to every node below it. The geographic area helper entity is an example.

Supportive entity
An entity that is used only to support the analytical data structures. The calendar date entity is an example.

Value group entity
An entity that groups together attributes used to perform analysis on a fact entity. Each attribute of a value group entity has a set of discrete values on which it is possible to aggregate the measures of a fact entity. The customer profile entity is an example.

Dimension relationship
A relationship between a fact entity and a fundamental entity or a classification entity from the data warehouse model.

Complex attribute
A measure that is calculated from other measures. The definition includes a formula of which the parameters are attributes.
Derived attribute
An attribute of which the value is derived from the value of one or more attributes from the data warehouse model. Typically this value includes fact entity measures that are not complex attributes. Their value is calculated from atomic data in the data warehouse model.

Other artifacts from the data warehouse model
The conformed dimensional model re-uses entities from the data warehouse model, typically to define its dimensions and its derived attributes. Refer to the artifacts of the enterprise model in Phase 3. Following are some of the artifacts:
- Fundamental entity
- Classification entity
- Supportive entity
- Basic attribute
- Relationship attribute

Customization of the conformed dimensional model

Typically, the customization performed on the analytical requirements in the context of the project scope drives the customization of the conformed dimensional model.

Thanks to the mappings to the analytical requirements, the customization of the conformed dimensional model is quite straightforward for all pre-existing Industry Model elements in analytical requirements. If a pre-existing element of analytical requirements is customized in the context of the scoped project, the customization should be reflected accordingly into the conformed dimensional model. For example, definitions and examples of a scoped customized measure need to be reviewed and customized in the conformed dimensional model.

For model elements that need to be created in the conformed dimensional model, you can find the description of this task in a special task description of the Industry Model Data Warehouse Development Method.

The following two sections offer examples of the rules to follow when you create new conformed dimensional model elements.

Rule 1

*If a new fact entity is added to the conformed dimensional model, the primary key needs to be added to the list of attributes (measures) originated from the analytical requirements.* This entity is composed of all relationship attributes that are the foreign keys on the dimension relationships. The verb phrases are either *for dimension* or *is dimension of.*

Notes:
- For a *snapshot* fact entity, the time dimension is replaced by a candidate key attribute, such as a reference date.
- The relationship attributes that compose the primary key should be named according to the dimension they represent. For example: Selling agent ID on a dimension relationship with the channel role fundamental entity.
- This will generate a primary key attribute in the EME Data Project Explorer that you should rename as the fact entity name suffixed with "PK"

- You need to add the following technical attributes to the primary key attribute:
  - A valid from date attribute, which represents the transaction time for the beginning of the time period during which the values of this recorded data are true in the source system. The valid from date attribute is also classified as a basic attribute, and it uses a data type timestamp of [TIMESTAMP].
  - A valid to date attribute, which represents the transaction time for the end of the time period during which the values of this recorded data are true in the source system. The valid to date attribute is also classified as a basic attribute, and it uses a data type timestamp of [TIMESTAMP].

- You need to map the new fact entity to the analytical requirements it supports.

**Rule 2**

*If a new dimension is added on a fact entity in the conformed dimensional model, the new dimension needs to be defined as a dimension relationship that links a fundamental or classification entity of the data warehouse model to the fact entity. The verb phrases are either for dimension or is dimension of.*

**Notes:**

- You should define the relationship attribute that represents the newly added dimension in the fact entity as a component of the primary key of the fact entity.
- The relationship attribute's name must reflect the dimension it represents. For example: Selling agent ID on a dimension relationship with the channel role fundamental entity.

Figure 16 shows a cutout of an example data diagram for a dimensional data model for a claim handling performance fact table. This fact table is a predefined fact table in the Claim Efficiency Analysis (CEA) application's conformed dimensional model. The attributes are omitted to improve readability.
The Claim Efficiency Analysis (CEA) application addresses the factors impacting the efficiency of the claims handling process by doing the following:

- Monitoring incoming recovery payments from third parties and re-insurers
- Assessing the distribution of claims amongst intermediaries and the loading of claim handlers
- Reconciling outstanding claims versus claims estimates
- Performing claims statistical analysis that might influence product development
- Analyzing the distribution of claims across all types of loss events.

The CEA solution enables the analytical fact table claim handling performance to monitor and identify inefficiencies in the claims handling process. Resulting reports help to optimize suppliers' networks and to improve operational efficiency and customer satisfaction.

### Summary of phase 4

This section described the conformed dimensional model and how to customize it. A conformed dimensional model can be transferred to the physical model using the InfoSphere Data Architect Wizard.

### Conclusion

This tutorial introduced the development process to build a stable and flexible data model for an insurance data warehouse. The approach used as business templates the industry-specific
concepts of the IBM Industry Model for Insurance Information Warehouse (IIW). The modeling itself is accomplished using the data modeling tool Information Data Architect (IDA) with the specific enhancement Enterprise Model Extender (EME). The appropriate design approach is described in the Data Warehousing Development Method (DWDM). You also read special hints about main artifacts of the development process. Understanding the description of the approach and the method prepared you to move through the major phases of this design process.

The method consists in general of four steps that moved from business objects and views to more and more technical objects. After you collected the requirements in phase 1, you built the business model in phase 2. The collection of the business requirements is done in cooperation with the business experts, who need to approve the resulting business model before you can continue the development process. In phases 1 and 2, you used business vocabulary and other EME-specific options of the IDA and EME tool environment. The tutorial explained the structure of these two conceptual models.

In phases 3 and 4, you built the logical data models (also called the enterprise model) from the business model. In phase 3, you built the logical model for the atomic part of the enterprise model, which is a normalized entity relationship (ER) model. In phase 4, you addressed a dimensional logical data model for data marts. This model is de-normalized and optimized for query and analysis performance. The sections that describe phases 3 and 4 offer the most important artifacts of the DWDM method and gave examples of how to customize some of these artifacts. These examples show best practice approaches to solving specific modeling tasks.

By following this clear roadmap of phases, you can deliver robust and stable data models that are also flexible enough to integrate new business requirements in the future.

This tutorial gave you a high-level understanding of Industry Models specifics. In future articles and tutorials on developerWorks, we will describe more about the IBM Industry Models. In particular, we plan to provide more detailed articles and tutorials on each of the four phases in the IIW data model development process with IDA and EME. We also plan to write an article that describes in detail the process of building a physical model out of an existing IIW logical model in the IDA and EME environment.
Related topics

- See "Scoping the IBM Industry Model for banking using Enterprise Model Extended and InfoSphere Data Architect" (developerWorks, Mar 2010) to learn how the process of scoping can yield business objects that will guide you to develop a meaningful logical data model.
- Read Exporting Industry Model Content to InfoSphere Business Glossary for more information.
- Refer to Mastering Data Warehouse Design: Relational and Dimensional Techniques, which provides an overview of exactly what is involved in designing and building a sustainable and extensible data warehouse.
- Find out more about InfoSphere Data Architect in the Integrated Data Management Information Center.
- Research more about InfoSphere Data Architect and grow your skills at the InfoSphere Data Architect page on developerWorks.
- Go to Data and process models for your industry on ibm.com to learn more.
- Get a trial version of InfoSphere Data Architect, and get familiar with its capabilities.

© Copyright IBM Corporation 2010
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)