Information Framework (SID)

User’s Guide

GB922 Addendum 1U

TM Forum Approved Version 8.6

 September, 2011

Notice

This material, including documents, code and models, has been through review cycles; however, due to the inherent complexity in the design and implementation of software and systems, no liability is accepted for any errors or omissions or for consequences of any use made of this material.

Under no circumstances will the TM Forum be liable for direct or indirect damages or any costs or losses resulting from the use of this specification. The risk of designing and implementing products in accordance with this specification is borne solely by the user of this specification.

This material bears the copyright of TM Forum and its use by members and non-members of TM Forum is governed by all of the terms and conditions of the Intellectual Property Rights Policy of the TM Forum (<http://www.tmforum.org/Bylaws/1094/home.html>) and may involve a claim of patent rights by one or more TM Forum members or by non-members of TM Forum.

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,

East Tower – 10th Floor,

Morristown, NJ  07960 USA

Tel No.  +1 973 944 5100

Fax No.  +1 973 944 5110

TM Forum Web Page: [www.tmforum.org](http://www.tmforum.org/)

Table of Contents

[Notice 2](#_Toc288580937)

[Table of Contents 3](#_Toc288580938)

[List of Figures 4](#_Toc288580939)

[1. Extending the Information Framework 5](#_Toc288580940)

[1.1. Introduction 5](#_Toc288580941)

[1.2. Creating a Model from the Information Framework Model 5](#_Toc288580942)

[1.3. Patterns for Extending Existing Business Entities 6](#_Toc288580943)

[1.3.1. Creating Packages to Hold Extensions 6](#_Toc288580944)

[1.3.2. Adding attributes 7](#_Toc288580945)

[1.3.3. Adding entities 9](#_Toc288580946)

[1.3.4. Adding associations 10](#_Toc288580947)

[1.4. Patterns For Defining New Aggregate Business Entities 11](#_Toc288580948)

[1.4.1. Business Entity Patterns 11](#_Toc288580949)

[1.4.2. Entity Specification/Entity Pattern 11](#_Toc288580950)

[1.4.3. Entity/Entity Role 13](#_Toc288580951)

[1.4.4. Composite/Component 14](#_Toc288580952)

[1.4.5. CharacteristicSpecification/CharacteristicValue 15](#_Toc288580953)

[1.4.6. Entity Classification Group 15](#_Toc288580954)

[1.5. Association, Attribute, and Package Naming Conventions 17](#_Toc288580955)

[1.5.1. Association Naming Conventions 17](#_Toc288580956)

[1.5.2. Attribute Naming Conventions 20](#_Toc288580957)

[1.5.3. Package Naming Conventions 21](#_Toc288580958)

[1.5.4. Guidelines for Naming Entities 22](#_Toc288580959)

[1.5.5. Guidelines for Defining Associations Between ABEs 22](#_Toc288580960)

[1.6. General Modeling Guidelines 22](#_Toc288580961)

[2. Using the Information Framework as an Integration Framework 24](#_Toc288580962)

[2.1 A Information Framework-Based XML Integration Framework 24](#_Toc288580963)

[3. Administrative Appendix 29](#_Toc288580964)

[3.1 About this document 29](#_Toc288580965)

[3.2 Document Life Cycle 29](#_Toc288580966)

[3.3 Document History 29](#_Toc288580967)

[3.3.1 Version History 29](#_Toc288580968)

[3.3.2 Release History 30](#_Toc288580969)

[3.4 Acknowledgments 31](#_Toc288580970)

List of Figures

[Figure U. 1 – Package Structure For Information Framework Extensions 6](#_Toc288580971)

[Figure U. 2 – Adding Attributes Using Sub-Classing 7](#_Toc288580972)

[Figure U. 3 – Adding Attributes Using Composition 8](#_Toc288580973)

[Figure U. 4 – Adding Attributes Using Stereotypes 8](#_Toc288580974)

[Figure U. 5 - Adding an Entity Related to an Existing Entity – Technique 1 9](#_Toc288580975)

[Figure U. 6 – Adding a Related Business Entity 10](#_Toc288580976)

[Figure U. 7 – Entity Specification/Entity Pattern 11](#_Toc288580977)

[Figure U. 8 – Service Specification/Service Use Of the Entity Spec/Entity Pattern 12](#_Toc288580978)

[Figure U. 9 – Service Specification ABE 12](#_Toc288580979)

[Figure U. 10 – Entity/Entity Role Pattern 13](#_Toc288580980)

[Figure U. 11 – Party/Party Role 13](#_Toc288580981)

[Figure U. 12 – Composite/Component Pattern 14](#_Toc288580982)

[Figure U. 13 – Product Offering Price Composite/Component 14](#_Toc288580983)

[Figure U. 14 – Entity Specification Characteristic/Entity Characteristic Pattern - 1 15](#_Toc288580984)

[Figure U. 15 – Entity Specification Characteristic/Entity Characteristic Pattern - 2 15](#_Toc288580985)

[Figure U. 16 – Classification Group Pattern 16](#_Toc288580986)

[Figure U. 17 – Market Statistic Use of Classification Group Pattern 17](#_Toc288580987)

[Figure U. 18 – Simple Association Naming 18](#_Toc288580988)

[Figure U. 19 - Using One Business Entity in the Name of an Association 19](#_Toc288580989)

[Figure U. 20 - Using Both Related Business Entities in the Name of an Association 20](#_Toc288580990)

[Figure U. 21 - Attribute Names Without Including Business Entity Name 21](#_Toc288580991)

[Figure U. 22 - Qualifying Attribute Names With the Business Entity Name 21](#_Toc288580992)

[Figure U. 23 - Using "ABE" in the Name of an ABE Package 22](#_Toc288580993)

[Figure U. 24 – Integration Framework Information Schema 25](#_Toc288580994)

[Figure U. 25 – XSD Representation of the Information Framework Customer ABE 26](#_Toc288580995)

[Figure U. 26 – Application Specific Extension to Customer Account Business Entity 26](#_Toc288580996)

[Figure U. 27 – Information Framework-Based Integration Framework 27](#_Toc288580997)

[Figure U. 28 – Inter-Application Communication Content Defined in Terms of Information Framework and Application Specific Extensions to the Information Framework 27](#_Toc288580998)

[Figure U. 29 – Integration Framework XSD File Structure 28](#_Toc288580999)

# 1. Extending the Information Framework

## Introduction

The Information Framework (SID) is a framework of frameworks. The framework of frameworks approach provides inherent extensibility and flexibility. The Information Framework program will be describing and documenting new functionality during each phase, so it is important to enable this new functionality to be easily added and extended without adversely affecting the overall Information Framework model.

This Addendum provides guidelines and examples of how to define extensions to the Information Framework. As a by-product, it also provides guidelines for developing ABEs[[1]](#footnote-1) that have not yet been modeled by the Information Framework team. Use of these guidelines will enable extensions made by different people and organizations to have the same structure, thus enabling these extensions to be compatible with each other as well as with the Information Framework itself. It also enables such extensions to be proposed back to the Information Framework team for official incorporation as part of the Information Framework.

The guidelines presented here include:

* Patterns and rules for extending existing business entities
* Patterns for new ABE development (which are used to design model packages and elements)
* Association, attribute, and package naming conventions
* General modeling guidelines.

Details about SID conformance can be found on the TM Forum Conformance Certification Assessment web pages.

The remainder of this document will use “Information Framework model” or “Information Framework” to refer to GB922 and its Addenda, which collectively define the Frameworx Business view of the Information Framework.

## Creating a Model from the Information Framework Model

**Note: A separate SID Model Administrator’s guide for the Rational Software Modeler UML 2.x too is currently under development. It will replace the prior content of this section, which focused on using Rational Rose, the previous UML 1.x tool used for the SID.**

## Patterns for Extending Existing Business Entities

These guidelines should be used when extending an existing ABE. By following them, the existing structure and content of an ABE will not be compromised. Additionally, if the Information Framework team makes changes to an ABE, adherence to these guidelines should minimize the impact to the extensions. The guidelines include:

creating packages to hold Information Framework extensions

adding attributes

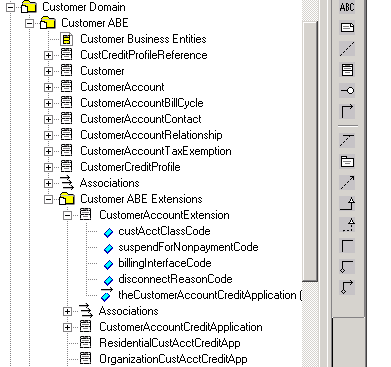
adding new entities

adding associations

Note that the above guidelines apply for extending business entities – a longer and stricter set of rules is being developed for extending system entities and will appear as a future Addendum to GB926.

### Creating Packages to Hold Extensions

UML packages should be used to hold extensions to the Information Framework model. There are a number of reasons for this. One reason is so that future versions of the Information Framework can be imported into a model without impacting extensions. Another reason is that it is easy to show the extensions made to the Information Framework model. Packages added should also be defined as control units so that they can be easily moved from model to model. The figure below shows an example of a packaged added to hold extensions to the Information Framework model Customer ABE.



Package to hold new extensions

Figure U. 1 – Package Structure For Information Framework Extensions

The package contains all new entities, attributes, and associations that are added to extend the Information Framework model.

### Adding attributes

There are four techniques that can be used to add attributes to an existing entity. If an entity is not sub-classed, then create a subclass of the Information Framework model business entity to which the attributes will be added. The subclass will inherit all of the attributes and associations from the Information Framework business entity thus maintaining the integrity of the Information Framework model. The new subclass holds the attributes as shown in the figure below.

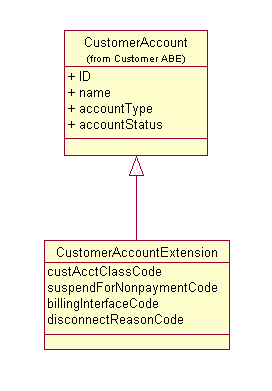


Figure U. 2 – Adding Attributes Using Sub-Classing

The name of the business entity that holds attribute extensions is an example. The actual name is at the discretion of the individual extending the Information Framework model. At a minimum, a consistent naming convention should be used.

Another technique can be used when the entity to be extended is already sub-classed. Here the attributes are “inherited” via the association, often referred to as an “IsA” association, implying the extension is a type of ServiceSpecification in the next figure.

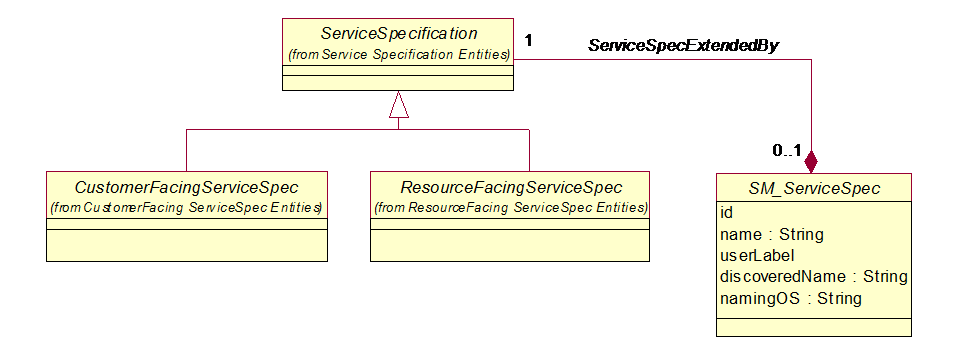


Figure U. – Adding Attributes Using Composition

The composition type of association is used to indicate that the life cycle of the framework entity and the entity containing the enterprise specific attributes share the same lifecycle. Where the composition appears in the association is up to the modeler. Here the composition is from the extension to the framework entity implying the key entity is the extension. If the modeler wishes to keep the framework entity as the key entity, then the composition would be from the framework entity. Some modelers prefer to use an aggregation type of association, which is also acceptable.

This is the most stable technique for adding attributes to an existing entity.

The third technique can be employed when there are a large number of extensions to make. For example, one member added extensions to half the framework’s entities, which would have increased the number of entities in the extended framework from about 1200 to 1800, increasing the apparent complexity of the model and making it more of a challenge to manage.

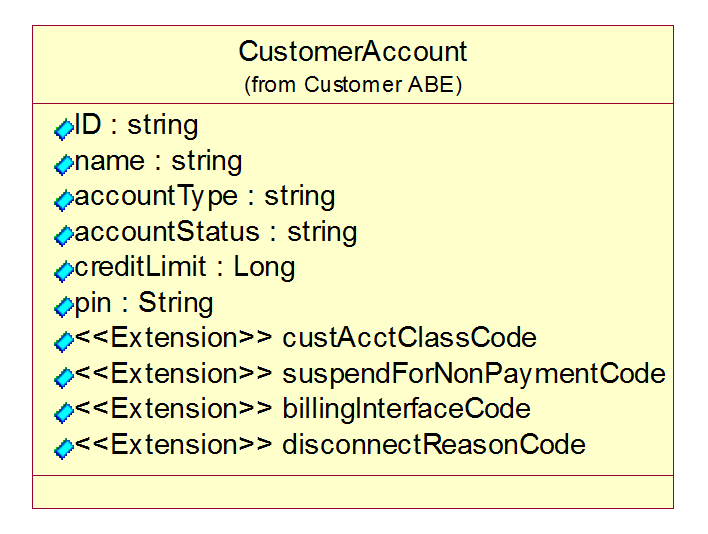


Figure U. – Adding Attributes Using Stereotypes

The disadvantage of this technique is that version control would have to be manually performed. For example, if a new version of CustomerAccount is introduced in the framework, either the extensions would have to be re-added or the new framework attributes would have to be manually added. Most framework entities are stable from their initial introduction, so the disadvantage of this technique rarely comes into play.

The fourth technique that can be employed is using the generalized CharacteristicSpecification/CharacteristicValue pattern that enables the dynamic addition of attributes.

### Adding entities

As in adding attributes, new entities should not be added directly to an existing Information Framework ABE package, but rather added to the package that holds the Information Framework model ABE extensions. Two techniques can be used to add entities to a Information Framework model ABE.

The first technique should be used when attributes as well as a new class need to be added to the Information Framework to extend an existing Information Framework class. In the example below, the new entity (CustomerAccountCreditApplication) needs to be related to an extension of the existing Information Framework CustomerAccout entity (CustomerAccountExtension). This is done by first subclassing the existing Information Framework entity (creating CustomerAccountExtension) and then defining an association (CreditApplicationSubmittedBy) between the subclass of the existing Information Framework entity and the new entity. The figure below shows an example of using this technique.



Figure U. 5 - Adding an Entity Related to an Existing Entity – Technique 1

The second technique can be used to add a related entity, such as CustomerAccountCreditApplication, when no attributes have been added to the existing related Information Framework model entity, such as CustomerAccount (there is no CustomerAccountExtension class). The figure below shows an example of using this technique.



Figure U. 6 – Adding a Related Business Entity

### Adding associations

The associations that exist among Information Framework model entities should not be changed or deleted. If a new association is required between existing entities, then they can be added. The association naming guidelines described in this addendum should be used.

## Patterns For Defining New Aggregate Business Entities

These guidelines should be used when developing a new ABE. Occasionally they may be used when adding more detail to an existing ABE. The guidelines include:

Business entity patterns

Association, attribute, and package naming conventions

Guidelines for naming entities

Guidelines for defining association classes vs. simple associations

### Business Entity Patterns

There are established sets of business entity patterns that are used in the Information Framework. These include:

Entity Specification/Entity

Entity/Entity Role

Composite/Component

Entity Characteristic Spec/Entity Characteristic

### Entity Specification/Entity Pattern

The Entity Specification/Entity pattern is used throughout the Information Framework model. Typically, most core business entities (that is, an entity within an ABE that is not dependent upon any other entity within the ABE, such as Customer, Product, Service) have their invariant attributes, methods, relationships and constraints defined by a specification, such as Product Specification and Service Specification. Customer does not have a related specification entity at this time.

This pattern should not be applied to existing ABEs, but should receive high consideration when adding a new ABE or detailing an existing ABE that has not been developed. The figure below shows the use of this pattern in the Root ABE.



Figure U. 7 – Entity Specification/Entity Pattern

The figure below shows the use of the Entity Specification/Entity pattern in the Service domain.

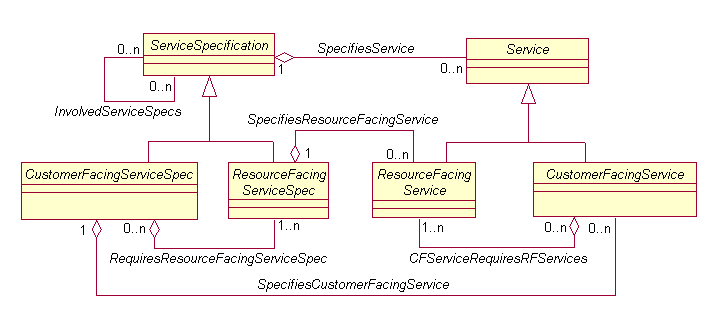


Figure U. 8 – Service Specification/Service Use Of the Entity Spec/Entity Pattern

Typically, there will be an ABE for the specification and an ABE for the entity characterized by the specification within each Information Framework domain or complex ABE. A complex ABE is one that decomposes into a number of other ABEs or into a number of nested ABEs. The reason for this is that each of these business concepts is complex enough to contain a number of related and dependent business entities. The figure below shows the Service Specification ABE to illustrate this point.



Figure U. 9 – Service Specification ABE

### Entity/Entity Role

Many business entities take on a variety of roles during their life of interest to a business. For example, an individual may be a customer and an employee of a service provider. The Entity/EntityRole pattern, shown in the figure below, is used throughout the Information Framework model to represent this concept.



Figure U. 10 – Entity/Entity Role Pattern

The figure below shows the use of this pattern in the Party ABE. Typically, each role that the entity plays is modeled as a subclass of the role entity. Many of these roles are also modeled as ABEs in their own right because of their importance to the business.



Figure U. 11 – Party/Party Role

### Composite/Component

Often, instances of a business entity are composed of other instances of the same business entity. For example, the price for cellular phone service may include a fixed monthly charge and a charge for excess minutes used. To model this, the Information Framework model employs the Composite/Component pattern, shown in the figure below.



Figure U. 12 – Composite/Component Pattern

The figure below shows the use of this pattern within the Product Offering Price ABE.



Figure U. 13 – Product Offering Price Composite/Component

### CharacteristicSpecification/CharacteristicValue

When constructing any model, it is almost impossible to discover all the possible attributes that characterize a business entity. Even if all the attributes can be found when the model is constructed, additional attributes will be found as a model is extended. Additionally, certain attributes characterize different types (represented by entity specifications) of business entities. For example, one product specification may be characterized by color and size, while another is characterized by bandwidth. The CharacteristicSpecification/CharacteristicValue pattern provides for this type of extensibility and characterization of different entities. The application of the generalized pattern is shown for the Product domain in the figures below.

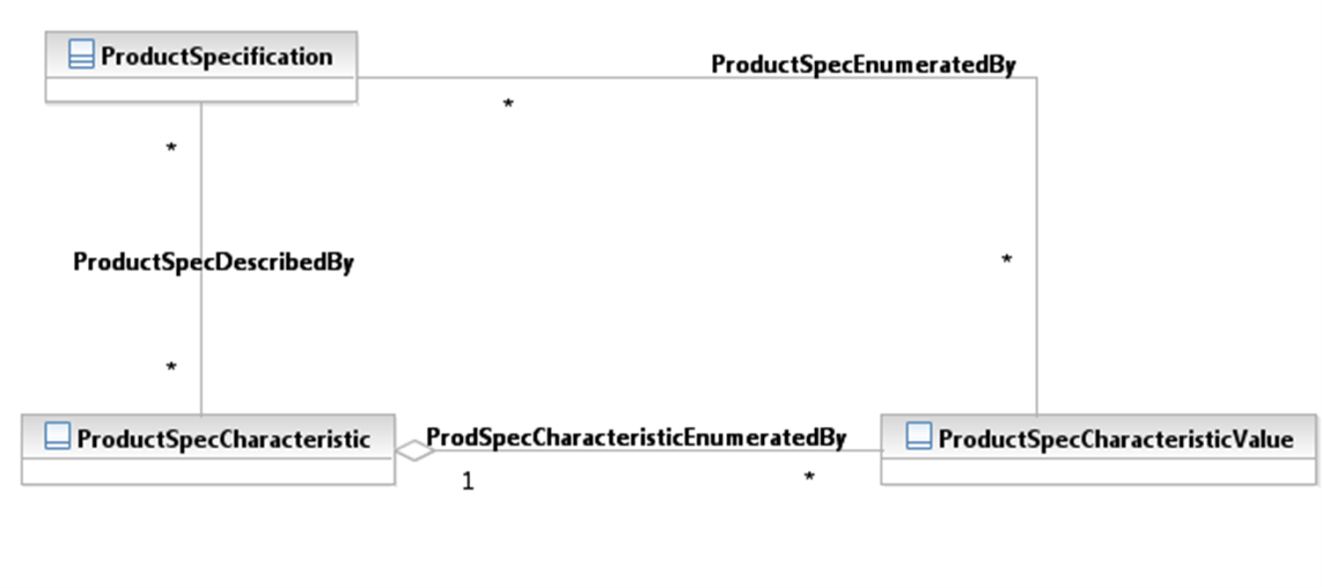


Figure U. 14 – Entity Specification Characteristic/Entity Characteristic Pattern - 1

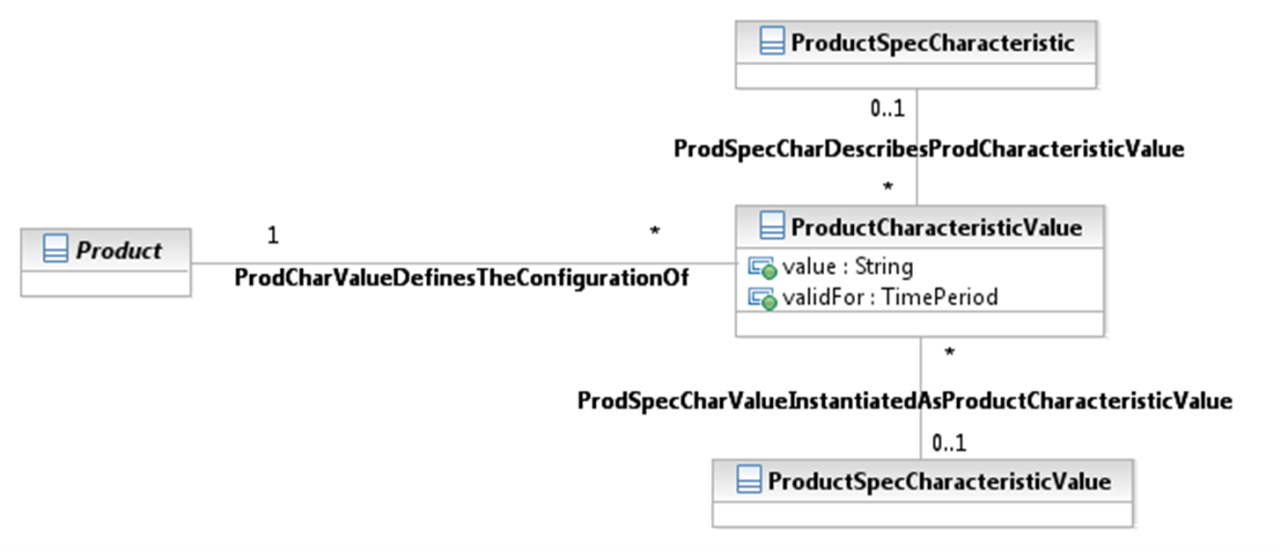


Figure U. – Entity Specification Characteristic/Entity Characteristic Pattern - 2

### Entity Classification Group

A Classification Group supports the grouping, organization and analysis of related business management and operational activities according to broad categories, subject areas, and so forth. The use of Classification groups, which are standardized and agreed across the organization allows for the identification and comparison of like activities, as well as the reduction of uncertainty, ambiguity and duplication.

Classification of information into agreed classification groups provides organizations a mechanism to aggregate, interrogate and manage business information on broad categories rather than on an instance or individual basis.

An additional rationale for grouping apart from the above is that it also allows identification of patterns and common behaviors across groups. It allows for "management on the large".

Examples of ClassificationGroups used in the telecommunications industry include: Market Segments, Product Portfolios, Distribution Channels, Consumer Revenue, Technology Platform, and so forth.

The figure below shows the ClassificationGroup pattern.



Figure U. 16 – Classification Group Pattern

The figure below shows the application of this pattern for the Market Statistic ABE.



Figure U. 17 – Market Statistic Use of Classification Group Pattern

## Association, Attribute, and Package Naming Conventions

### Association Naming Conventions

One method of naming associations is to simply use a verb. To interpret the meaning of the name, the two related business entities are used as subject and object. For example, in the figure below the association between ProductSpecification and BusinessInteractionItem is interpreted as ProductSpecification Involvedin BusinessInteractionItem; the meaning is interpreted by using the upper business entity as the subject. Similarly, the association between Service and BusinessInteractionItem is interpreted as Service InvolvedIn BusinessInteractionItem; the meaning is interpreted from left to right.

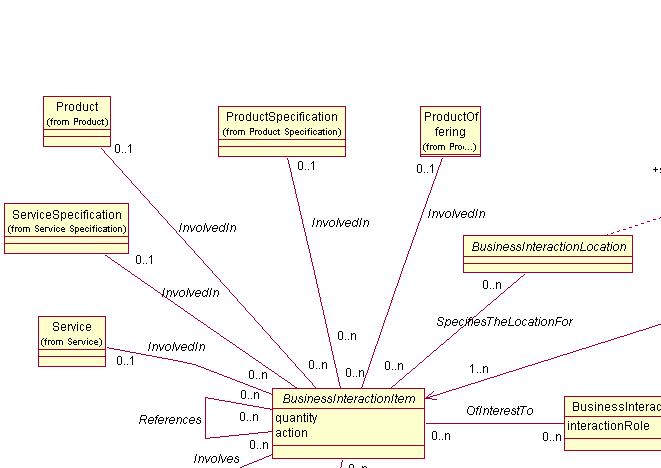


Figure U. 18 – Simple Association Naming

This method of interpretation works well when the business entities retain their relative positions in every diagram in which they appear. However, if in another diagram the positions of business entities are reversed, then the interpretation of the associations is incorrect. For example, if the positions of BusinessInteractionItem and ProductSpecification are reversed, the interpretation of the association is BusinessInteractionItem InvolvedIn ProductSpecification, which is an incorrect interpretation.

This problem with interpreting the meaning of an association can be eliminated by introducing the name of either or both of the related business entities into the name of the association. If one business entity name is used, the name should include the subject as shown in Figure U. 19 - Using One Business Entity in the Name of an Association below. For example, BusinessInteractionTypeCategorizes (BusinessInteraction, the implied object). This naming convention is always used unless the association name results in a duplicate name.

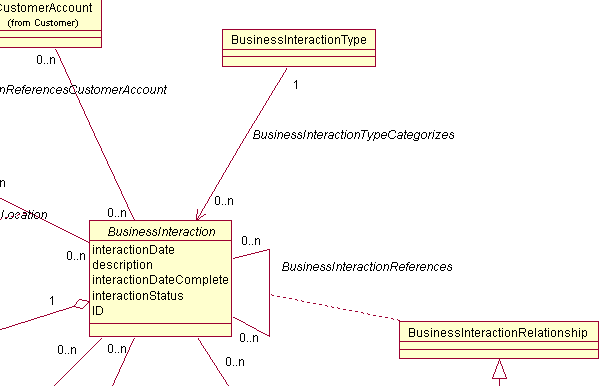


Figure U. 19 - Using One Business Entity in the Name of an Association

To resolve duplicate association names both related business entities are used to form the name of the association as shown in Figure U. 20 below.

For example, BusinessInteractionItemInvolvesProductSpecification. The subject of the association name should be the ABE that owns the association. This means that in the model the association appears in the package containing the business entity that appears as the subject in the association name. In this example, the association appears in the BusinessInteraction package, in which BusinessInteractionItem resides.

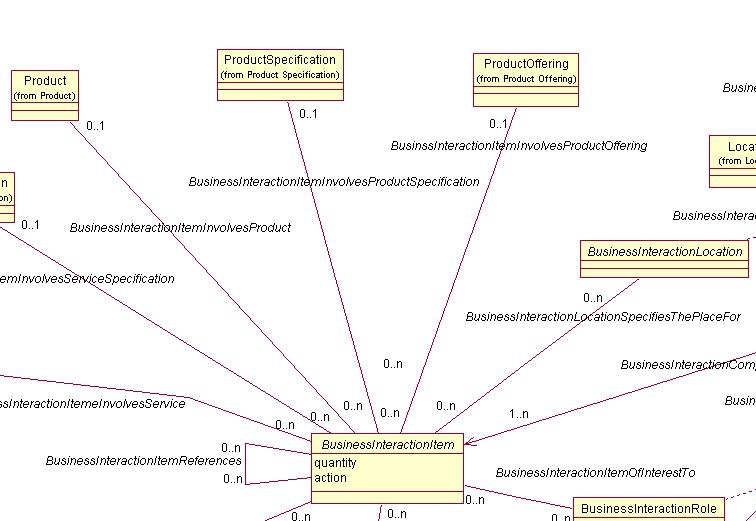


Figure U. 20 - Using Both Related Business Entities in the Name of an Association

It may appear that always using both related business entities in the name of the association would be more prudent. One convention is much simpler to use and understand than two. However, there are a number of reasons for employing two naming conventions. They are:

1. Using both related business entities results in long association names that tend to clutter up diagrams and are difficult to use, and
2. Using the subject/verb/object naming convention will most likely only be needed on a minimum number of occasions.

### Attribute Naming Conventions

Naming attributes, like naming associations, can include the name of the attribute’s business entity in the name. This is exemplified the figures in this section below.

When an attribute is unique across all business entities or when an attribute is shared among business entities, but maintains the same semantics (characteristics) among the business entities, the name does not include the name of the business entity. An example is shown in the figure below.

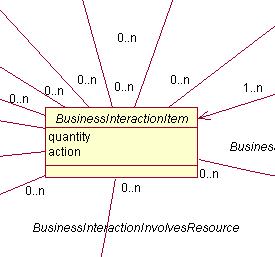


Figure U. 21 - Attribute Names Without Including Business Entity Name

When an attribute, such as interactionStatus, in the figure below does possess different semantics across business entities of which the attribute is a characteristic, then the attribute name should be qualified with the name of the business entity. In the figure below, interactionStatus can take on a number of different values dependent upon the business entity that it characterizes.

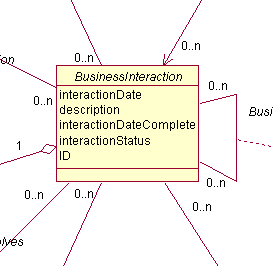


Figure U. 22 - Qualifying Attribute Names With the Business Entity Name

In some cases, a modeler may choose to qualify all attributes that characterize a business entity. This is at the discretion of the modeler.

In the case of Boolean attributes, the recommended convention is to use “is” or “has” as a prefix. For example, the attribute “active” which can take on the values of “yes” or “no”, should be named isActive.

### Package Naming Conventions

Package names in the Information Framework model correspond to the names of ABEs in the Information Framework.

There are no naming conventions are recommended other than using “ABE” as the suffix in the package name of ABEs. Package names employing the ABE suffix are shown in the figure below.

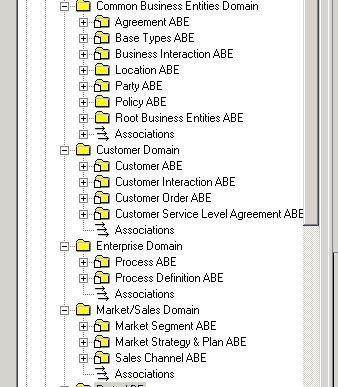


Figure U. 23 - Using "ABE" in the Name of an ABE Package

### Guidelines for Naming Entities

Business entities should take on a name that will be familiar to a business individual. The name should not be expressed using technical jargon.

### Guidelines for Defining Associations Between ABEs

In general, inter-ABE associations should have a multiplicity of 0…n - 0...n. This is done in order to minimize coupling (dependencies) between ABEs that reside in different domains.

## General Modeling Guidelines

Any work performed on the Information Framework model should follow best practice modeling guidelines, such as:

* Avoiding the use of multiple inheritance
* Minimal use of association
* When considering the use of an association class to model an entity, a class should be used instead when
  + The entity will inherit from another entity
  + The entity will be sub-classed
  + The entity will be related to other entities not involved in the association
* Describing each artifact is required
* Following the GB922 series of documents with regard to format and content

# 2. Using the Information Framework as an Integration Framework

One challenge faced when integrating applications is reconciling different terms used for the same concepts in the applications being integrated. A second challenge is reconciling the structure of each application’s information model. Being a common vocabulary and an information model framework makes the Information Framework an ideal candidate to be used to meet these integration challenges.

## A Information Framework-Based XML Integration Framework

Defining the Information Framework in terms of XML schemas, or XSDs, is an approach that can be used to create an integration framework. Messages that are interchanged between applications are defined using the Information Framework model components (business entities, attributes, and their associations) along with application specific extensions to the Information Framework.

Since the Information Framework model is a framework, applications are expected to extend the Information Framework model. These extensions complete the definition of business entities that are to be used by an application. Extensions can include objects such as additional business entities, attributes, and/or associations. The Information Framework model and extensions to the Information Framework form the basis of the XML information schema as shown in the figure below.



Figure U. 24 – Integration Framework Information Schema

The figure below shows the Information Framework model’s Customer ABE as an XML schema.

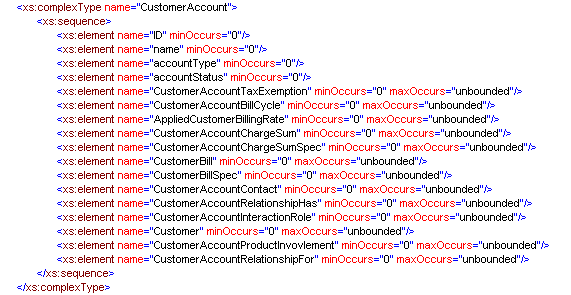


Figure U. 25 – XSD Representation of the Information Framework Customer ABE

The next figure shows an example of an application specific extension to the Information Framework Customer Account business entity within the Customer ABE.

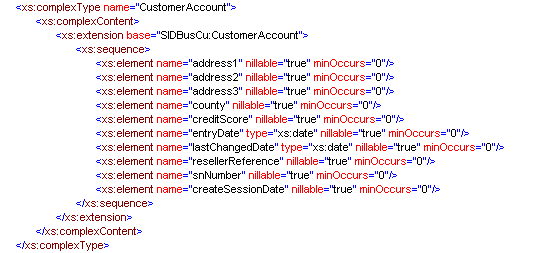


Figure U. 26 – Application Specific Extension to Customer Account Business Entity

This type of application specific extension is used if the application contains all the Information Framework attributes and associations. If the application contains a subset of the Information Framework attributes and associations, then a restriction base is used for the extension. Use of this method is described in GB922 Addendum X – Information Framework XML Schema (XSD) Overview.

These two sets of information schema form the basis from which inter-application communication is constructed. The figure below shows how an application-specific schema can be constructed using the Information Framework as a foundation.



Figure U. 27 – Information Framework-Based Integration Framework

The next figure shows the structure of an XSD, defined in terms of the Information Framework-based integration framework, in graphical form.

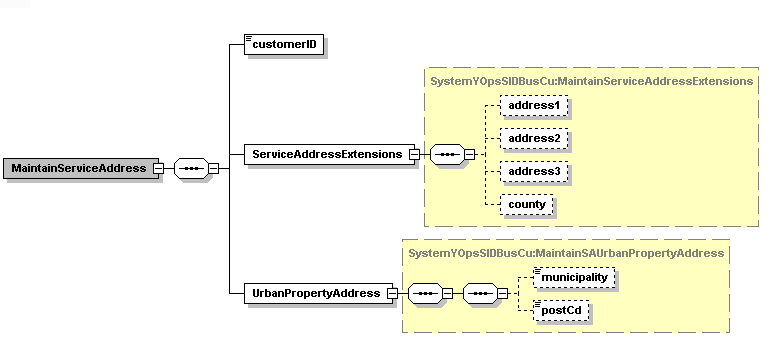


Figure U. 28 – Inter-Application Communication Content Defined in Terms of Information Framework and Application Specific Extensions to the Information Framework

Information Framework domains can be used to organize the schema for each component of the integration framework as shown in the figure below. The figure shows the Information Framework Customer domain XSD, an application operation message payload XSD for the Customer domain, and a application specific extension XSD for the Customer domain.

XSD File Naming

Figure U. 29 – Integration Framework XSD File Structure

1. Administrative Appendix

This Appendix provides additional background material about the TeleManagement Forum and this document. In general. sections may be included or omitted as desired, however a Document History must always be included..

## 3.1 About this document

This is a TM Forum Guidebook. The guidebook format is used when:

The document lays out a ‘core’ part of TM Forum’s approach to automating business processes. Such guidebooks would include the Telecom Operations Map and the Technology Integration Map, but not the detailed specifications that are developed in support of the approach.

Information about TM Forum policy, or goals or programs is provided, such as the Strategic Plan or Operating Plan.

Information about the marketplace is provided, as in the report on the size of the OSS market.

## 3.2 Document Life Cycle

This document is being issued for Member Evaluation. The purpose of an Evaluation Version is to encourage input based on experience of members and the public as they begin to use the document. Following the Evaluation Period, documents that are seen to deliver value are candidates for formal approval by the TM Forum. All documents approved by the TM Forum undergo a formal review and approval process.

This document will continue under formal change control. Supporting work will be issued as companions to this document. A document of this type is a “living document,” capturing and communicating current knowledge and practices. Further inputs will be made because of detailed work ongoing in the TM Forum and the industry.

## 3.3 Document History

### 3.3.1 Version History

<This section records the changes between this and the previous document version as it is edited by the team concerned. Note: this is an incremental number which does not have to match the release number>

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Date Modified** | **Modified by:** | **Description of changes** |
| 0.1 | Sept 2003 |  | First Draft |
| 0.2 | Sept 2003 |  | Second Draft based on comments from sub-team (John Strassner and Josh Salomon) |
| 0.3 | Oct 2003 |  | Additional comments from sub-team |
| 0.4 | Dec 2003 |  | Final review from sub-team, based on other Phase IV deliverable |
| 0.5 | Dec 2003 |  | Updates based on review comments by Cliff Faurer and Wayne Sigley |
| .6 | Feb 2004 | John Reilly | Updated based on SMT review |
| 6.0 | July 2005 | John Reilly | Updated based on member contributions. |
| 6.1 | November 2005 | Tina O'Sullivan | Converted to new template and corrected various administrative items. |
| 6.2 | November 2005 | Tina O'Sullivan | Figure label |
| 6.3 | November 2005 | Tina O'Sullivan | Moved fig 5. |
| 6.4 |  | Tina O'Sullivan | Updated notice statement & document status |
| 6.5 | May 2009 | Alicja Kawecki | Minor updates to reflect TM Form Approved Status |
| 8.1 | March 2010 | Pascale Pecha | edits |
| 8.2 | March 2010 | Alicja Kawecki | Minor cosmetic corrections for web posting |
| 8.3 | June 2010 | Alicja Kawecki | Updated Notice |
| 8.4 | December 2010 | John Reilly | Updates based on change requests |
| 8.5 | March 2011 | Alicja Kawecki | Minor formatting corrections prior to web posting and ME |
| 8.6 | September 2011 | Alicja Kawecki | Updated to reflect TM Forum Approved status |

### 3.3.2 Release History

<This section records the changes between this and the previous Official document release>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Release Number** | | **Date Modified** | **Modified by:** | | **Description of changes** | |
| Release 6.0 | | 31-Oct-2005 | J. Reilly | |  | |
| 8.1 | January,2010 | | | Ken Dilbeck | | Naming changes per Marketing |
| 9.5 | December 2010 | | | John Reilly/John Wilmes | | Updates based on change requests |

## 3.4 Acknowledgments

This document was prepared by the members of the TeleManagement Forum Information Framework team:

The Shared Information/Data Model is a genuinely collaborative effort. The TM Forum would like to thank the following people for contributing their time and expertise to the production of this document. It is just not possible to recognize all the organizations and individuals that have contributed or influenced the introduction. We apologize to any person or organization we inadvertently missed in these acknowledgments.

Key individuals that reviewed, provided input, managed, and determined how to utilize inputs coming from all over the world, and really made this document happen were:

|  |  |
| --- | --- |
| **Name** | **Affiliation** |
| Ian Best | TeleManagement Forum |
| John Reilly | MetaSolv Software |
| Wayne Sigley | Telstra |
| John Strassner | Motorola |
| Josh Salomon | Amdocs |
| Greg Fidler | Practical Enterprise Architecture P/L |

1. An ABE is an Aggregated Business Entity – a collection of common classes that model a set of related concepts. Please see GB922, Concepts and Principles, for more information. [↑](#footnote-ref-1)