International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

M.3190

(07/2008)

SERIES M: TELECOMMUNICATION MANAGEMENT, INCLUDING TMN AND NETWORK MAINTENANCE

Telecommunications management network

Shared information and data model (SID)

Recommendation ITU-T M.3190

ITU-T M-SERIES RECOMMENDATIONS

TELECOMMUNICATION MANAGEMENT, INCLUDING TMN AND NETWORK MAINTENANCE

Introduction and general principles of maintenance and maintenance organization	M.10–M.299
International transmission systems	M.300–M.559
International telephone circuits	M.560–M.759
Common channel signalling systems	M.760–M.799
International telegraph systems and phototelegraph transmission	M.800–M.899
International leased group and supergroup links	M.900–M.999
International leased circuits	M.1000–M.1099
Mobile telecommunication systems and services	M.1100–M.1199
International public telephone network	M.1200–M.1299
International data transmission systems	M.1300–M.1399
Designations and information exchange	M.1400–M.1999
International transport network	M.2000–M.2999
Telecommunications management network	**M.3000–M.3599**
Integrated services digital networks	M.3600–M.3999
Common channel signalling systems	M.4000–M.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T M.3190

Shared information and data model (SID)

Summary

Recommendation ITU-T M.3190 is an introduction to the TM Forum SID analysis model and a specification through reference of the ITU-T subset of SID (the product and service business entities). The SID analysis model scope covers the information required to implement use cases based on the eTOM processes. The SID analysis model is implementation independent, focusing on what the information is and what its relationships are, not on how these are to be implemented.

The SID applies primarily to service providers' businesses and engaged stakeholders, e.g., system integrators, independent software vendors and network equipment providers.

Source

Recommendation ITU-T M.3190 was approved on 14 July 2008 by ITU-T Study Group 4 (2005-2008) under Recommendation ITU-T A.8 procedure.

Keywords

Data model, information model, methodology, TM Forum, UML.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Recommendation ITU-T M.3190

Shared information and data model (SID)

1 Scope

This Recommendation is an introduction to the TM Forum SID analysis model and a specification through reference of the ITU-T subset of SID. The SID analysis model scope covers all of the information required to implement use cases based on the eTOM processes [ITU-T M.3050.x]; in effect, this means that the SID covers a large proportion of the information needs for a service provider. However, the SID is still being developed and while the documents available today cover a substantial part of a service provider's core information needs, they do not cover all of them.

The SID analysis model is implementation independent, focusing on *what* the information is and *what* its relationships are, not on *how* these are to be implemented.

The SID applies primarily to service providers' businesses and engaged stakeholders: system integrators (SI), independent software vendors (ISV) and network equipment providers (NEP).

Parts of the SID may also apply directly to, or be adaptable to, other non-Telco service provider enterprise businesses.

The focus of this Recommendation is the product and service business entities.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T M.3020] Recommendation ITU-T M.3020 (2007), *Management interface specification methodology*. <http://web.itu.int/rec/T-REC-M.3020/>

[ITU-T M.3050.x] Recommendations ITU-T M.3050.x-series (2007), *Enhanced Telecom Operations Map (eTOM)*. <http://web.itu.int/rec/T-REC-M.3050.0/>

[ITU-T M.3100] Recommendation ITU-T M.3100 (2005), *Generic network information model*. <http://web.itu.int/rec/T-REC-M.3100/>

[TMF GB922] TM Forum GB922, *Shared Information/Data (SID) Model. Concepts, Principles, and Domains, Release 7.5.* <http://www.tmforum.org/DocumentsInformation/GB922InformationFramework/35040/article.html>

[TMF GB922A3] TM Forum GB922, *Shared Information/Data (SID) Model. Addendum 3 – Product Business Entity Definitions, Release 7.5.* <http://www.tmforum.org/DocumentsInformation/GB922InformationFramework/35040/article.html>

[TMF GB922A4SO] TM Forum GB922, *Shared Information/Data (SID) Model. Addendum 4SO – Service Overview Business Entity Definitions, Release 7.5.* <http://www.tmforum.org/DocumentsInformation/GB922InformationFramework/35040/article.html>

[OMG UML] OMG, *Unified Modelling Language Specification, Version 1.5*. <http://www.uml.org/>

Additional non-normative references can be found in the Bibliography.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 aggregate business entity [TMF GB922]: An Aggregate Business Entity (ABE) is a well-defined set of information and operations that characterize a highly cohesive, loosely coupled set of business entities.

3.1.2 attribute [TMF GB922]: An attribute is a fact that describes a business entity.

3.1.3 business entity [TMF GB922]: A Business Entity represents something of interest to the business that may be tangible things (such as a Customer), active things (such as a Customer Order), or conceptual things (such as a Customer Account). Business entities are characterized by attributes and participate in relationships with other business entities. Business entity instances typically move through a well-defined life cycle.

3.1.4 domain [TMF GB922]: A Domain is a collection of Aggregate Business Entities associated with a specific management area. Domains that make up the SID Framework are consistent with eTOM level 0 concepts.

3.1.5 relationship [TMF GB922]: A relationship is an association of business interest between two business entities, or between a business entity and itself.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 information model: An information model for a problem space represents all relevant things, i.e., objects in that problem space along with all of their relevant attributes, and relationships.

– There may be a number of different relevant viewpoints represented for different purposes.

An information model is *independent* of implementation, i.e., independent of platform, language, protocol, etc.

– For example, it does not represent whether it is defined using CORBA IDL or a XML schema.

Not all things (objects) in the information model will result in implementation forms.

– For example, an information model view may be specified for the purpose of communication across management interfaces. However, not all things (objects, attributes, relationships) in that model will be visible across a management interface.

– The things to be encoded in an implementation form will be identified in the information model.

3.2.2 data model: A data model is an encoded form of relevant parts of the information model for a *specific* platform, language and/or protocol, i.e., this is an implementation view.

– A data model adds no problem space specific properties, it only applies encoding and augments the information model with encoding directives.

– The relevant parts to be encoded for a specific application will be identified in the information model.

A single information model may give rise to one or more data models.

- Data models are typically designed with respect to the characteristics of the platform, language or protocol.
- For example, an information model may be represented via a data model defined in CORBA IDL and also a data model defined using a XML schema.

4 Abbreviations

This Recommendation uses the following abbreviations:

ABE	Aggregate Business Entity
CMIP	Common Management Information Protocol
eTOM	Enhanced Telecom Operations Map
MISM	Management Interface Specification Methodology (see [ITU-T M.3020])
NgOSS	New Generation Operations Systems and Software
QoS	Quality of Service
SID	Shared Information and Data model
SLA	Service Level Agreement
UML	Unified Modelling Language
VPN	Virtual Private Network

5 Conventions

SID is described using UML [OMG UML] as described in Appendix III.

An analysis pattern describes a problem which occurs over and over again, and describes the core of the solution to that problem. Key analysis patterns that have been applied for the modelling of SID are described in Appendix I.

6 SID introduction

6.1 SID as an information model

6.1.1 What it is

A common information model will streamline the processes associated with information exchange, both within an enterprise and between the enterprise and its external stakeholders.

Having a standard information model provides the following benefits:

- it allows for simplification of information management, by providing a common terminology and reducing unnecessary variation;
- it allows for unification of information both within an enterprise and between enterprises;
- it provides a bridge between the business and information technology groups by providing definitions that are understandable by the business, but are rigorous enough to be used for software development.

These benefits then enable business benefits relating to cost, quality, timeliness and adaptability of enterprise operations, allowing an enterprise to focus on value creation for their customers.

An information model should model fundamental concepts and not be affected by minor business environment changes like organization restructures.

The information model should be an enabler of change (new products, processes and ways of doing business), not a blocker to progress.

An information model (as opposed to a data model) is an abstraction and only provides a high level view of things of interest to the business. It aids in understanding the scope and breadth of the business, rather than the depth.

An information model is a way of representing and structuring information that has advantages over other common artifacts, as described in Table 1.

Table 1 – Information model comparison to other artifacts

Artefact	Artefact limitations	Information model advantage
Glossary	Hard to understand how the concepts relate	An information model links concepts together and provides a visual overview.
Descriptive document	Often rambling & inconsistent with overlapping terminologies used	An information model is more precise, using the power of UML to ensure that the information is represented in a consistent manner.
Database model or source code	Too detailed to get a feel of the big picture. Hard to match to business requirements. Hard to convert from a particular language/platform/environment.	A database model is a solution, but it may not map to your problem. A business is likely to have many database models and these need to be understood against a common view of the business (e.g., for EAI integration, business intelligence, etc.). The SID documents a unifying view: the basis for decisions, not just the results.

There are many views as to what an information model is and does. For the SID, the initial work has focused on creating a conceptual or analysis model, (also known as a domain model [b-Patterns-1] [b-Patterns-2] or a semantic model [b-Zachman]). The model is a representation of real world objects in which there is interest from the business.

An analysis model includes:

– things in which the business is interested (domain entities);

– how they are related to one another (associations);

– key details about those things which help to define them unambiguously (domain-level attributes).

An information model is useful:

– as a starting point for internal modelling work, applications and messages between software components or database schemas;

– to help in defining a common business terminology, e.g., for integration activities;

– to help in business transformation programs;

– to help in understanding business concepts and their relationships;

– as a source of inspiration – for a new view on traditional practices.

6.1.2 What it is not

An information model – and SID – is not:

– a database model;

– a definition of how software will be written;

– an implementation of software classes;

– a software API (application programming interface);

– a set of component messaging definitions.

The analysis model will, however, provide business definitions that should be a major input to the definition of the above items.

The SID analysis model is not:

– a definition of software classes;

– enough to do an implementation on its own;

– a definition of platforms, protocols, software languages or software products to be used for developing software components.

The SID is not a "magic bullet" but should be seen as a key component of an integrated information architecture (model framework + methodology) within an overall enterprise architecture.

Other parts of an information architecture are typically:

– Information policy – Directs information management within a business.

– Communication policy – Information maintenance and publishing standards (model downloads, data glossary/dictionary definitions, interface definitions, data translations, etc.).

– Governance processes – Managing separate component models within the overarching framework principles including interface design; managing data definitions within process design; managing data definition within applications design.

To achieve maximum business benefits, businesses need to consider how to integrate the use of the SID into their existing processes and procedures, especially those relating to IT definition & development.

6.2 Document structure

The SID is organized in a number of Addenda. The following documents are normative parts of this Recommendation.

Reference	Title	Included aggregate business entities
GB922 [TMF GB922]	**Business View Concepts, Principles, and Domains** describing the organizing structure of the SID Business View shared information/data, the relationship to other TMF deliverables, and views of the contents of the model.	
GB922 Addendum 3 [TMF GB922A3]	**Product**	Product Specification, Product Offering, Product, Product Offering Price
GB922 Addendum 4SO [TMF GB922A4SO]	**Service**	Service Specification, Service

NOTE – Only the service and product ABEs [TMF GB922A3] [TMF GB922A4SO] are normatively referenced by this Recommendation. The complete set of ABEs can be found in Appendix II.

The SID Addenda use a mixture of descriptive text, UML diagrams and tables which formally define the entities and attributes. This provides concise and precise definitions with background reasoning and explanations.

6.3 Relationship with other Recommendations

6.3.1 Relationship with [ITU-T M.3020]

The management interface specification methodology (MISM) defined by [ITU-T M.3020] specifies a three-phase process with features that allow traceability across the three phases. The three phases apply industry-accepted techniques using object-oriented analysis and design principles. The three phases are requirements, analysis and design.

The requirements and analysis phases produce UML specifications. The design phase uses network management paradigm specific notation. The outputs of the 3 phases are:

– Requirements phase – Requirements.

– Analysis phase – Implementation-independent specification.

– Design phase – Technology-specific specification.

The SID as a conceptual model is an implementation-independent specification. Given the historic background, the SID has not been developed based on the methodology of [ITU-T M.3020], but the result maps well to the outcome of the analysis phase. A distinction between the SID and [ITU-T M.3020] is that the SID is limited to the conceptual information model and without the interface (operations and notifications) aspects also covered by [ITU-T M.3020]. However, SID can be used as an input to the specification of concrete interfaces based on the methodology.

6.3.2 Relationship with eTOM ([ITU-T M.3050.x])

The SID and eTOM [ITU-T M.3050.x] model the same subject matter from different points of view. The eTOM focuses on the business processes; the dynamic aspects of the business.

The SID analysis model focuses on the static aspects of the business; the things that the business processes act on, their characteristics and relationships.

The SID analysis model is organized using the SID framework.

The top level of grouping, called a SID domain, aligns with the eTOM level 0 definitions of:

– Market/Sales

– Product

– Customer

– Service

– Resource

– Supplier/Partner.

The framework also represents common business entities (e.g., party, location, policy) and has a placeholder domain for Enterprise entities which have not yet been fully defined (e.g., Financial information).

The SID framework is extensively documented in the main GB922 document [TMF GB922] with an overview given in clause 7.

6.3.3 Relationship with [ITU-T M.3100]

[ITU-T M.3100] is a generic information model focused on modelling of equipment and network resources. [ITU-T M.3100] is structured into fragments:

- Network fragment
- Managed element fragment
- Physical equipment fragment
- Logical equipment fragment

- Termination point fragment – Network element view
- Termination point fragment – Network view
- Topology and connectivity fragment – Network view
- Telemetry fragment
- Transmission fragment
- Cross-connection fragment
- Functional area fragment

The products and service domains defined by SID can be considered as complements to the current content of [ITU-T M.3100], i.e., there is no direct overlap between the two models.

The current [ITU-T M.3100] is based on the GDMO notation. As a result, [ITU-T M.3100] can be considered a data model (see definitions). GDMO is a flexible notation that need not be supported by CMIP. The result is that [ITU-T M.3100] today is a combination of an information model (the conceptual aspects) and a specific data model for use with CMIP.

6.3.4 Relationship with [b-ITU-T M.1400.x]

The [b-ITU-T M.1400.x] series Recommendations focus on the specification of the terminology and grammar as perceived by the end users. The specification requirements may include requirements to support end-user interaction at their human-computer interfaces.

A detailed description of the relationship and differences can be found in Appendix V.

7 SID Framework

This clause describes the SID framework that is used to organize the SID model. The SID framework provides a high-level view into the SID entities, as well as providing an organizing structure in which the SID business entities reside. Additional details can be found in [TMF GB922].

Figure 1 below shows how the domains contained within the SID framework align with eTOM level one domains/concepts. Whether taking a process or information perspective, it is important to be viewing the same set of concepts. The alignment is also a necessary enabler when mapping eTOM processes to SID business entities.

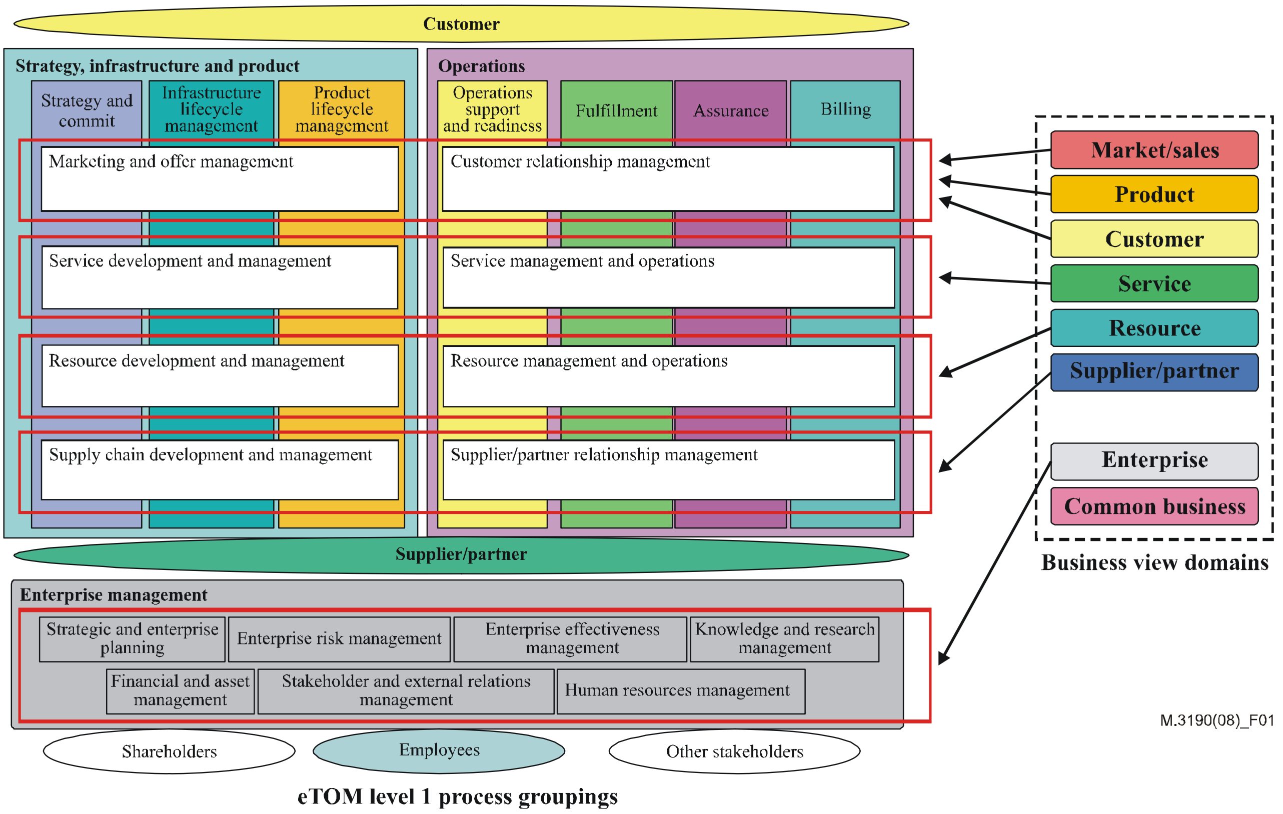
Customer
Strategy, infrastructure and product
Strategy and commit
Infrastructure lifecycle management
Product lifecycle management
Marketing and offer management
Service development and management
Resource development and management
Supply chain development and management
Operations
Operations support and readiness
Fulfillment
Assurance
Billing
Customer relationship management
Service management and operations
Resource management and operations
Supplier/partner relationship management
Supplier/partner
Enterprise management
Strategic and enterprise planning
Enterprise risk management
Enterprise effectiveness management
Knowledge and research management
Financial and asset management
Stakeholder and external relations management
Human resources management
Shareholders
Employees
Other stakeholders
eTOM level 1 process groupings
Market/sales
Product
Customer
Service
Resource
Supplier/partner
Enterprise
Common business
Business view domains
M.3190(08)_F01

Figure 1 – eTOM/SID concepts/domains

7.1 SID framework – Level 1 ABEs

Figure 2 shows the SID framework. The framework depicts the domains and level 1 aggregate business entities (ABE) contained within each domain.

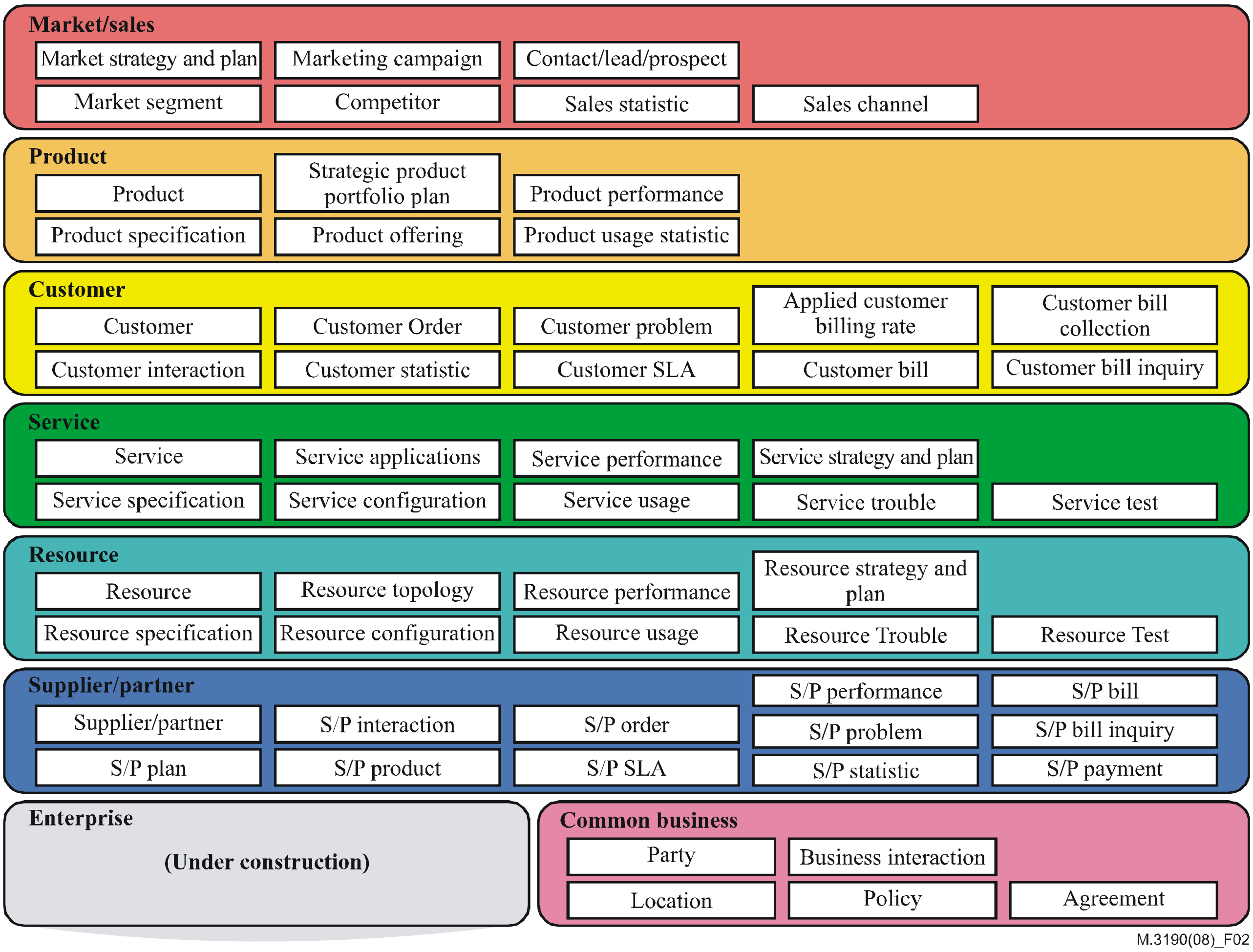

Figure 2 – SID business view framework

As the development of the SID model has progresed, subsequent levels of ABEs have been identified.

7.2 ABE categorization

The ABE content and structure within each domain should be consistent. To ensure this, each ABE is aligned with a categorization pattern as described below. The pattern can also be used to confirm the completeness of each domain's ABE.

ABE categories include:

- Strategy and plan
- Managed entity
- Managed entity specification – A description of a ManagedEntity that might allow it to be built.
- Interaction – A communication with a ManagedEntity. This is a type of BusinessInteraction.
- Configuration – The internal structure of a ManagedEntity.
- Performance – The measure of ManagedEntity quality.
- Test – A means of interrogating a ManagedEntity in order to understand its state(s).
- Trouble – A problem associated with a ManagedEntity. Alarms, outages and faults are examples.

– Price – The cost of a ManagedEntity.

– Usage – A period of time during which a ManagedEntity is in use.

NOTE – ABEs in market/sales and common business domains have not yet been categorized.

8 Product domain

This clause is an introduction to the product domain; detailed specifications can be found in [TMF GB922A3].

The product domain is concerned with the life cycle of products and information and contract operations related to the products' life cycle. The domain contains aggregate business entities that deal with the strategic portfolio plans, products offered, product performance, product usage statistics, as well as the product instances delivered to a customer.

Table 2 – Relationship of product ABEs to eTom processes

Product ABEs	Primary vertical eTOM process groupings	Primary eTOM Level 2 processes	Secondary eTOM Level 2 processes
Strategic product portfolio plan Is concerned with the plans of the product portfolio, which product offerings to make available to each market segment and the plans to development and deploy product offerings, as well as retirement of products.	M&OM – SC	Product & offer portfolio planning	Market strategy & policy
			Product & offer development & retirement
Product specification Defines the functionality and characteristics of product offerings made available to the market.	M&OM – PLM	Product & offer development & retirement	Product & offer capability delivery
			Service configuration & activation
			SM&O support & readiness
Product offering Represents tangible and intangible goods and services made available for a certain price to the market in the form of product catalogues. This ABE is also responsible for targeting market segments based on the appropriate market strategy.	M&OM – PLM	Product & offer development & retirement	Marketing fulfillment response
			Product marketing communications & promotion
			Product & offer capability delivery
			Selling
			Order handling

Table 2 – Relationship of product ABEs to eTom processes

Product ABEs	Primary vertical eTOM process groupings	Primary eTOM Level 2 processes	Secondary eTOM Level 2 processes
Product Represents an instance of a product offering subscribed to by a party, such as a customer, the place where the product is in use, as well as configuration characteristics, such as assigned telephone numbers and internet addresses. The Product ABE also tracks the services and/or resources through which the product is realized.	CRM – F	Order handling	Selling
			Service configuration & activation
			Problem handling
			Customer QoS/SLA management
			Billing & collections management
			Service problem management
			Service quality management
			Service & specific instance rating
			SM&O support & readiness
Product usage statistic Represents usage trends of products associated with various demographics, such as market segment.	CRM – OSR	CRM support & readiness	Product & offer development & retirement
			Problem handling
			Customer QoS/SLA management
			Enterprise performance assessment

Table 2 – Relationship of product ABEs to eTom processes

Product ABEs	Primary vertical eTOM process groupings	Primary eTOM Level 2 processes	Secondary eTOM Level 2 processes
Product performance The product performance ABE handles product performance goals, the results of end-to-end product performance assessments, and the comparison of assessments against goals. The results may include the identification of potential capacity issues.	CRM – OSR	CRM support & readiness	Product & offer development & retirement
			Customer QoS/SLA management
			Service quality management
			Resource performance management
			Enterprise performance assessment

9 Service domain

This clause is an introduction to the service domain; detailed specifications can be found in [TMF GB922A4SO].

The service domain consists of a set of layered ABEs that are used to manage the definition, development, and operational aspects of services provided by a management system. Entities in this domain support various eTOM processes that deal with the definition, development and management of services offered by an enterprise. This includes agreement on service levels to be offered, deployment and configuration of services, management of problems in service installation, deployment, usage, or performance, quality analysis, and rating. Finally, this domain also includes entities to perform planning for future offerings, service enhancement or retirement, and capacity.

Table 3 – Relationship of service ABEs to eTom processes

Service ABEs	Primary vertical eTOM process groupings	Primary eTOM level 2 processes	Secondary eTOM level 2 processes
Service specification The service specification ABE contains entities that define the invariant characteristics and behaviour of both types of service entities. This enables multiple instances to be derived from a single specification entity. In this derivation, each instance will use the invariant characteristics and behaviour defined in its associated template. Entities in this ABE focus on adherence to standards, distinguishing features of a service, dependencies (both physical and logical, as well as on other services), quality, and cost. In general, entities in this ABE enable services to be bound to products and run using resources.	SD&M – PLM	Service development & retirement	Service capability delivery
			Service configuration & activation
			Service problem management
			Service quality management
			Service & specific instance rating
			SM&O support & readiness
Service The service ABE contains entities that are used to represent both customer-facing and resource-facing types of services. Entities in this ABE provide different views to examine, analyze, configure, monitor and repair services of all types. Entities in this ABE are derived from service specification entities.	SM&O – F	Service configuration & activation	Service capability delivery
			Service problem management
			Service development & retirement
			SM&O support & readiness
			Service quality management
			Service & specific instance rating

Table 3 – Relationship of service ABEs to eTom processes

Service ABEs	Primary vertical eTOM process groupings	Primary eTOM level 2 processes	Secondary eTOM level 2 processes
Service applications The service applications ABE contains entities that define different types of services implemented as applications. There are at least three different types of entities in this domain, which form a hierarchy of applications. The simplest is a coordinated set of service mechanisms, such as the ability to fine-tune QoS. The next level of application is a type of transport, such as a VPN, that uses one or more service mechanisms to provide basic functionality to a PartyRole. The highest level of application is a management application that uses transport(s) and mechanisms to provide an end-to-end manageable application, such as distance learning or VoIP.	SM&O – F	Service configuration & activation	Service problem management
			Service capability delivery
			Service development & retirement
			Service quality management
			Service & specific instance rating
Service configuration The service configuration ABE contains entities that are used to represent and manage configurations of CustomerFacingService and ResourceFacingService entities. This set of entities also provides details on how the configuration of each of these types of services can be changed. The entities in this ABE depend on entities in the resource domain, which provide the physical and logical infrastructure for implementing a service. They all define dependencies between a higher-level service and any sub-services that are used by the higher-level service.	SM&O – F	Service configuration & activation	Service capability delivery
			SM&O support & readiness
			Resource provisioning
Service performance The service performance ABE collects, correlates, consolidates, and validates various performance statistics and other operational characteristics of customer and resource facing service entities. It provides a set of entities that can monitor and report on performance. Each of these entities also conducts network performance assessment against planned goals, performs various aspects of trend analysis, including error rate and cause analysis and service degradation. Entities in this ABE also manage the traffic generated by a service, as well as traffic trend analysis. This is important for newer technologies that separate data, control and management functions for a given service.	SM&O – A	Service quality management	Service problem management
			SM&O support & readiness
			Service development & retirement
			Resource performance management

Table 3 – Relationship of service ABEs to eTom processes

Service ABEs	Primary vertical eTOM process groupings	Primary eTOM level 2 processes	Secondary eTOM level 2 processes
Service test The service test ABE contains entities that are used to test customer and resource facing service entities. These entities are usually invoked during installation, as a part of trouble diagnosis, or after trouble repair has been completed.	SM&O – F	Service configuration & activation	Service problem management Service quality management
Service trouble The service trouble ABE manages faults, alarms, and outages from a service point-of-view. This is then correlated to trouble tickets, regardless of whether the cause is physical or logical. Other entities in this ABE are used to direct the recovery from each of these three types of problems. They provide the ability to associate resource faults and alarms to degradation and outages of services that run on those resources. These functions are independent of the resources and technologies used to build the service. A third set of entities in this ABE is used to differentiate between customer-reported problems and network-induced problems.	SM&O – A	Service problem management	SM&O support & readiness Problem handling Service quality management Resource trouble management
Service usage The service usage ABE collects service consumption data, and generates service usage records, for use by other business entities. The entities in this ABE provide physical, logical, and network usage information.	SM&O – B	Service & specific instance rating	Resource data collection & processing Service quality management Service development & retirement
Service strategy & plan The service strategy and plan ABE contains entities that are used to address the need for enhanced or new services, as well as the retirement of existing services, by the enterprise. These entities have a strong dependency to both entities in the resource and product domains. Resulting efforts, such as deciding what resources to use to host a service, or what services are used to support new product specifications, are also supported, as are service demand forecasts.	SD&M – SC	Service strategy & planning	Product & offer portfolio planning Service capability delivery Service development & retirement Resource strategy & planning

Appendix I

Analysis patterns used in SID

(This appendix does not form an integral part of this Recommendation)

NOTE – The content of this appendix is based on the SID primer [b-TMF GB922 primer].

This appendix introduces patterns that have emerged as we modelled the eTOM processes in the SID analysis model. We use these patterns to improve the quality of the SID model, in particular coupling & cohesion [b-Design].

Since the primary focus of this primer is not on patterns, only a brief overview will be given and a reference given to more detailed definitions.

A pattern can be described as follows [b-Language]:

"Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice".

I.1 Specification

The specification pattern [b-Patterns-2] is used in the SID when:

"There needs to be a description about an item or service, independent of the current existence of any examples of those items or services".

The specification entity does not represent a concept or thing, but *information about* a concept or thing. Specification entities are found in business concepts such as catalogs, manufacturing specifications, recipes and other documentation that relate to types of things rather than individual instances.

This pattern is used in the SID Product Addendum (*Product & ProductSpecification Entities*) and most other SID addenda.

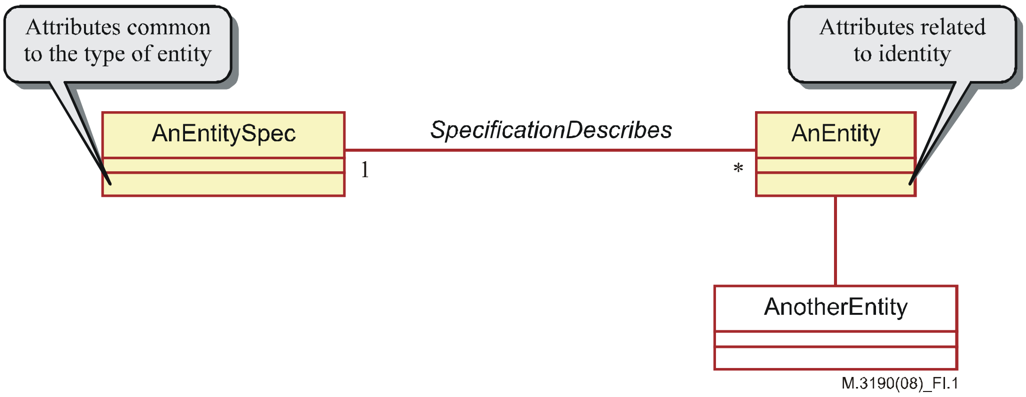

Figure I.1 – Specification pattern

I.2 Abstract superclass

We use the abstract superclass pattern [b-Abstract] to add superclass entities to allow us to group similar entities together. We make the parent entity abstract to show that it will not occur in the real world, but is a modelling construct.

The abstract superclass pattern helps make the SID more general and easier to extend. This pattern helps us improve the modularity (coupling & cohesion [b-Design]) of the SID model.

This pattern is used extensively in the SID (e.g., the *Party Entity* is an abstract superclass of the *Individual Entity* and the *Organization Entity*).

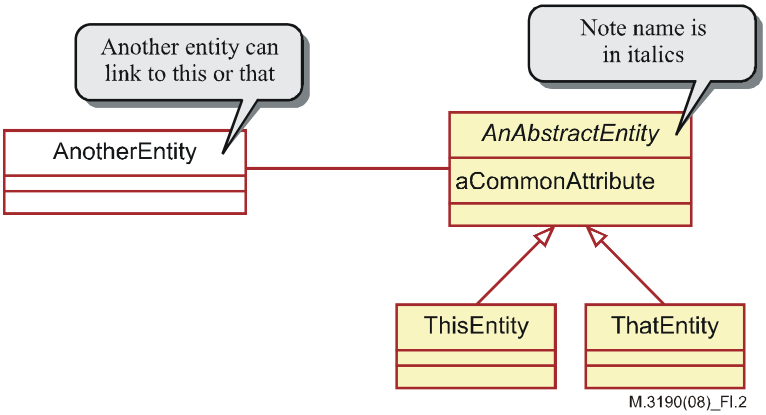

Figure I.2 – Abstract superclass pattern

We will now go through a simple example of how the abstract entity pattern can be used to improve a model.

We may be asked to model a piece of business information: "We sell bookshelves, chairs and desks to charities, individuals, companies and government departments". If we just model this directly, then we will produce a model that will be complex and hard to extend. If we add two new abstract entities, we can now say: "We sell furniture to BusinessParties". The new concepts show a more fundamental understanding of the business. Note that Furniture is a placeholder for the "Concrete Entities" Chair, Desk & Bookshelf. Furniture is abstract because when we sell something to a particular customer we do not say: "I just sold a furniture to John Brown", instead we use one of the specific (concrete) entities.

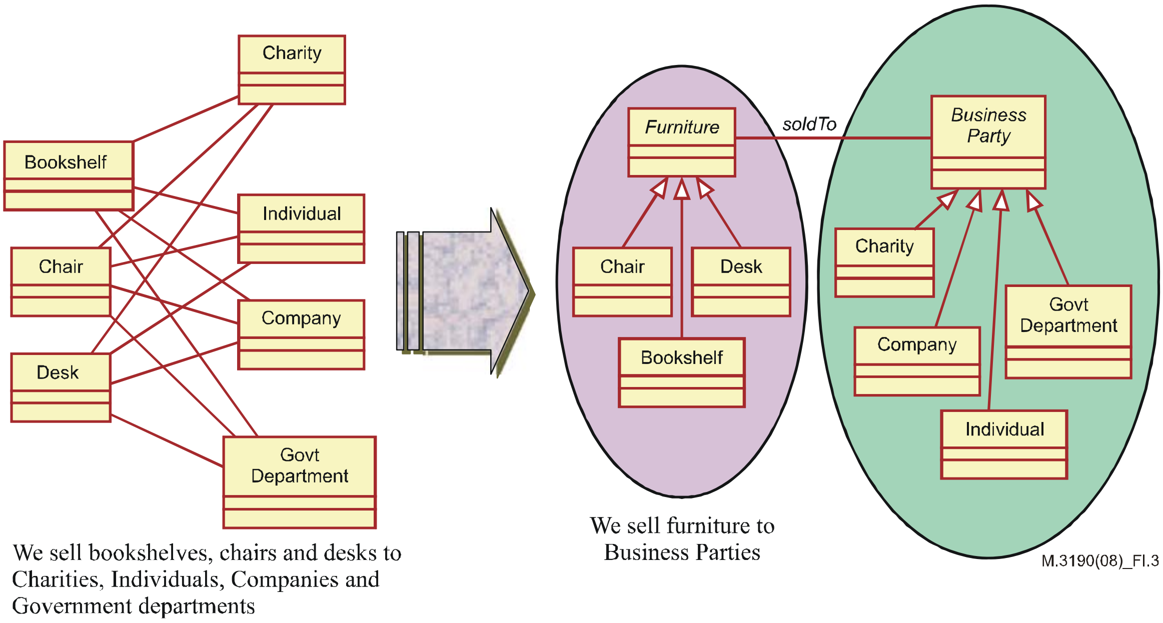

Figure I.3 – Abstract superclass pattern example

I.3 Composite

The composite pattern [b-Patterns-3] is usually thought of as a design pattern [b-IBM]:

"Assemble objects into tree structures. COMPOSITE simplifies clients by letting them treat individual objects and assemblies of objects uniformly."

In the SID, we use the composite pattern when there is a business concept where a single thing or a collection of those things can be used interchangeably. For example, in a warehouse, the concept of a "stock item" may include parts, sub-assemblies and complete items.

This pattern is used extensively in the SID (e.g., *Party*, *Individual & Organization Entities*).

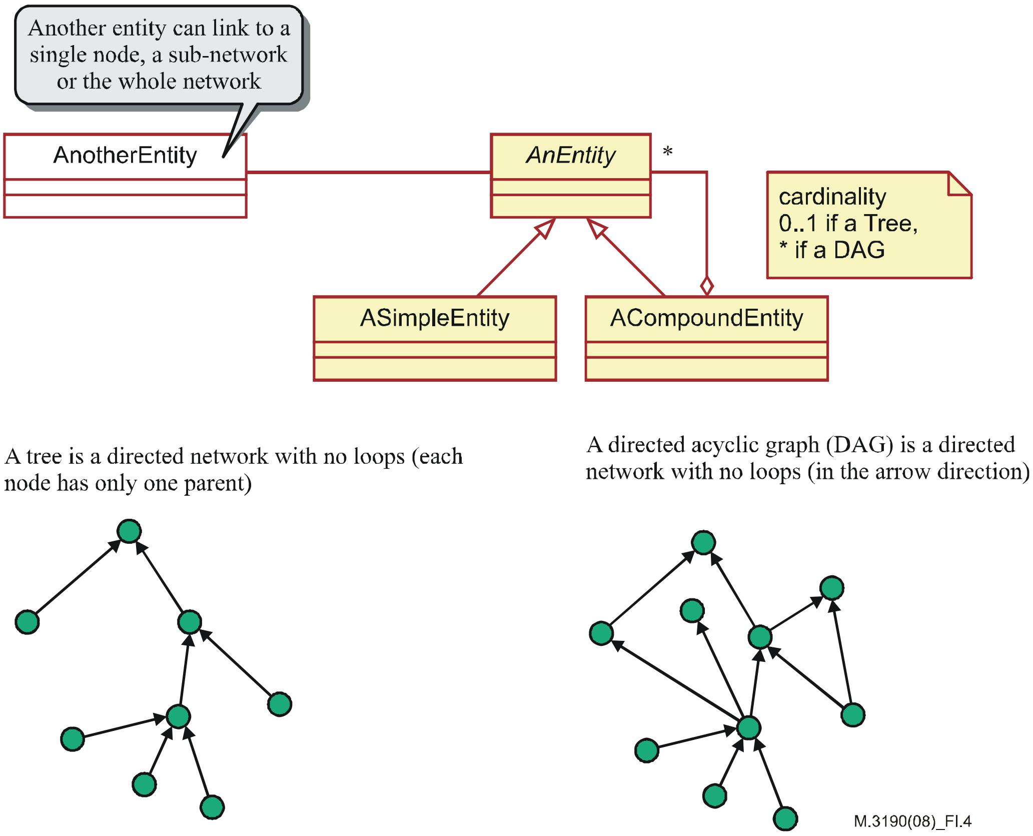
Another entity can link to a single node, a sub-network or the whole network
AnotherEntity
AnEntity
*
cardinality 0..1 if a Tree, * if a DAG
ASimpleEntity
ACompoundEntity
A tree is a directed network with no loops (each node has only one parent)
A directed acyclic graph (DAG) is a directed network with no loops (in the arrow direction)
M.3190(08)_FI.4

Figure I.4 – Composite pattern

Note that when we form composites of physical things, these often form tree structures. Composites of logical things and composites of specifications often form directed acyclic graphs. In effect, this is just common sense. If there is a specific hard drive, it can only be plugged into a single computer at a time. A *type* of hard drive, however, may be able to be plugged into many different types of computers (PCs, Macintoshes, sun workstations, etc.).

I.4 Role entity

The role entity pattern [b-Roles] allows us to represent behaviour with respect to a given context, e.g., "a person who is a witness in the context of a legal trial".

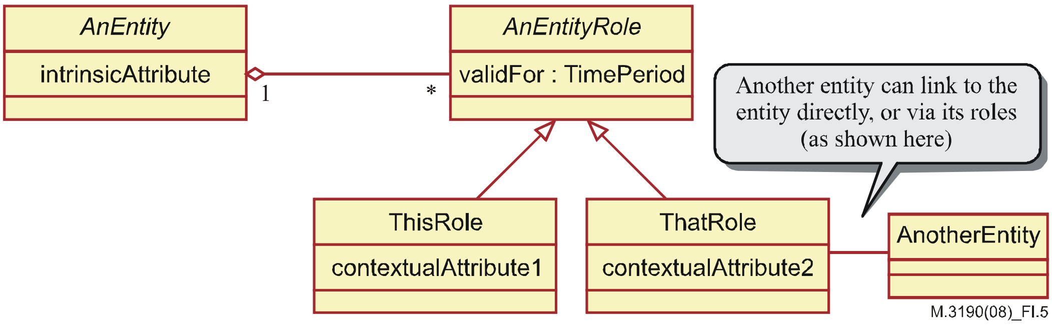

Figure I.5 – Role entity pattern

Use of roles is a fundamental pattern that helps us simplify a model, and make it more closely represent the real world.

Intrinsic attributes are those that a thing always has. Contextual attributes are those that relate to a thing in certain situations.

When modelling a thing, question whether it has different behaviours in different circumstances, i.e., how it is being used, not just what it is. If so, the role pattern may be useful.

Representing the concept and its roles using two entities makes the SID model more robust to change and reduces duplication.

I.5 Temporal state entity

We use the temporal state entity pattern [b-Patterns-3] when we wish to be able to show the states of an entity, the attributes for each state and the temporal or life cycle aspects of an entity. An entity's state will change over time and we may wish to keep only the current state or a complete history, depending on the business requirements.

Separating the characteristics that we need to monitor over time into a separate entity allows us to show this more clearly, than if it was shown as attributes in the entity.

Note that this pattern looks similar to the "Role Entity" pattern but represents a different concept.

The articles "Patterns for things that change with time" on Martin Fowler's [b-Patterns-1] website give a good understanding of time-related issues and are well worth reading.

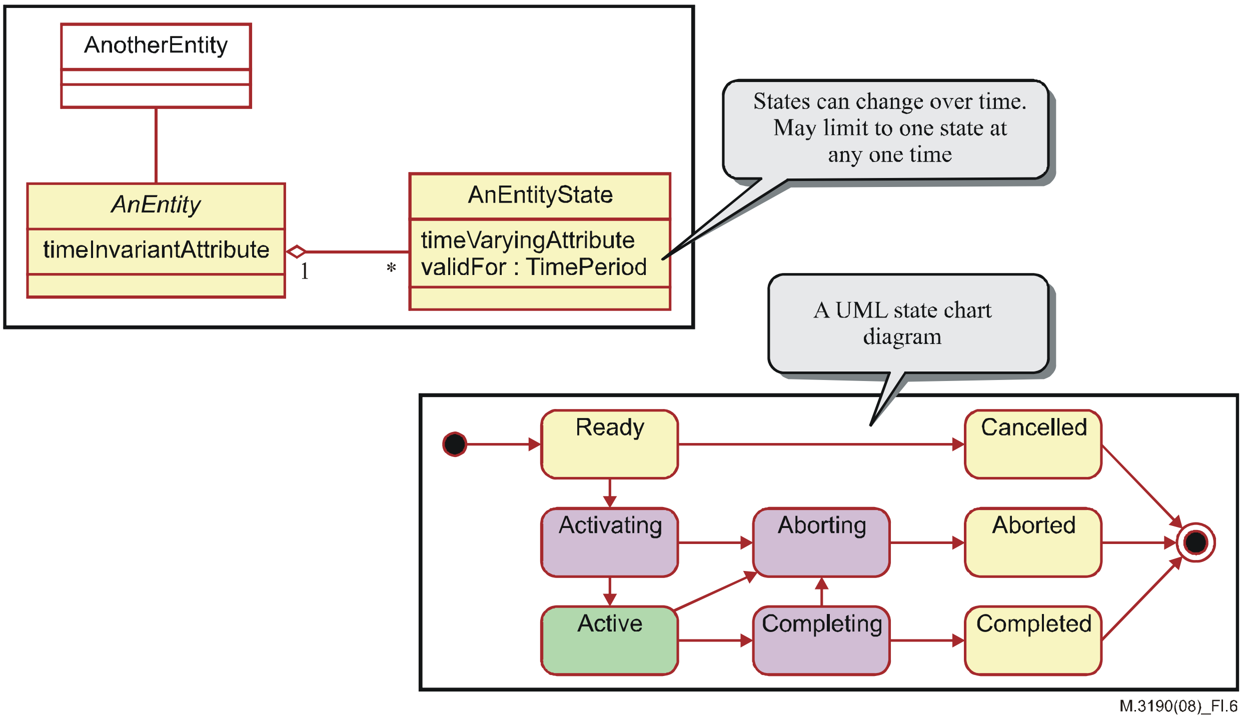

Figure I.6 – Temporal state entity pattern

I.6 Self-relationship

The self-relationship pattern is [b-ISO 19108] used when an instance of an entity may have a relationship to other instances of the same entity. For instance, a family tree could be formed by linking individuals to their parents.

This pattern is used in the SID Project analysis model (e.g., *Project Element Entity*).

Table I.1 – Self-relationship types

Relationship type	Description
Dependency	This is where the two entities have some starting or finishing dependency, e.g., Activity 2 cannot start until Activity 1 is complete.
Succession	This is where one or more entities are replaced by one or more other entities. This is an abstract relationship and one of the concrete types listed below must be used.
Substitution	This is a one for one replacement, e.g., Activity 1 is no longer valid and has been replaced by Activity 1A.
Division	This is a one for many replacement, e.g., Activity 1 is no longer valid and has been replaced by Activities 1A, 1B and 1C.
Fusion	This is a many for one replacement, e.g., Activities 1A, 1B and 1C are no longer valid and have been replaced by Activity 1Z.

Note how the model uses an association entity, allowing us to define attributes for the association.

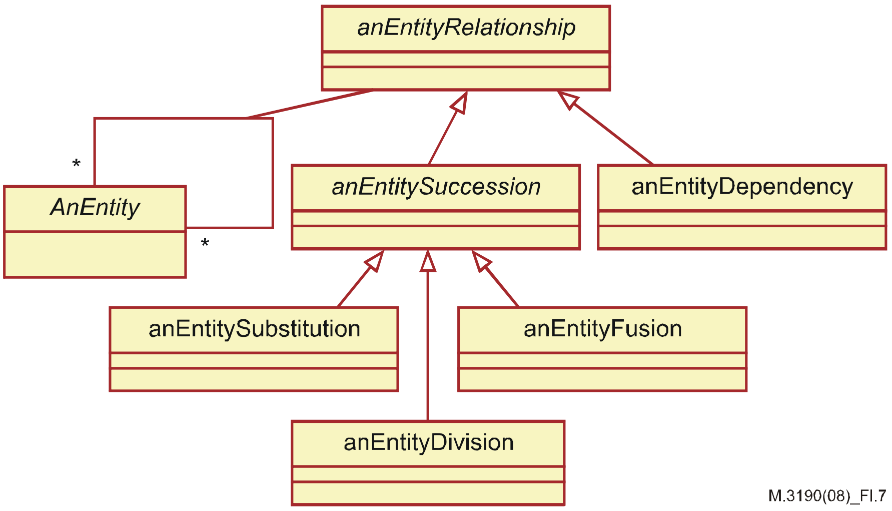

Figure I.7 – Self-relationship pattern

I.7 Entity specification characteristic/entity characteristic

When constructing any model, it is almost impossible to discover all the possible attributes, such as configurable and/or static properties and features, which 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 of business entities, represented by entity specifications in the SID.

For example, an instance of a mobile device product specification may be characterized by configurable colour and static size, while hi-speed Internet is characterized by configurable bandwidth. The EntitySpecificationCharacteristic/EntityCharacteristic pattern provides this type of extensibility and characterization for different instances of specifications and their related entity instances. The pattern is shown in Figure I.8.

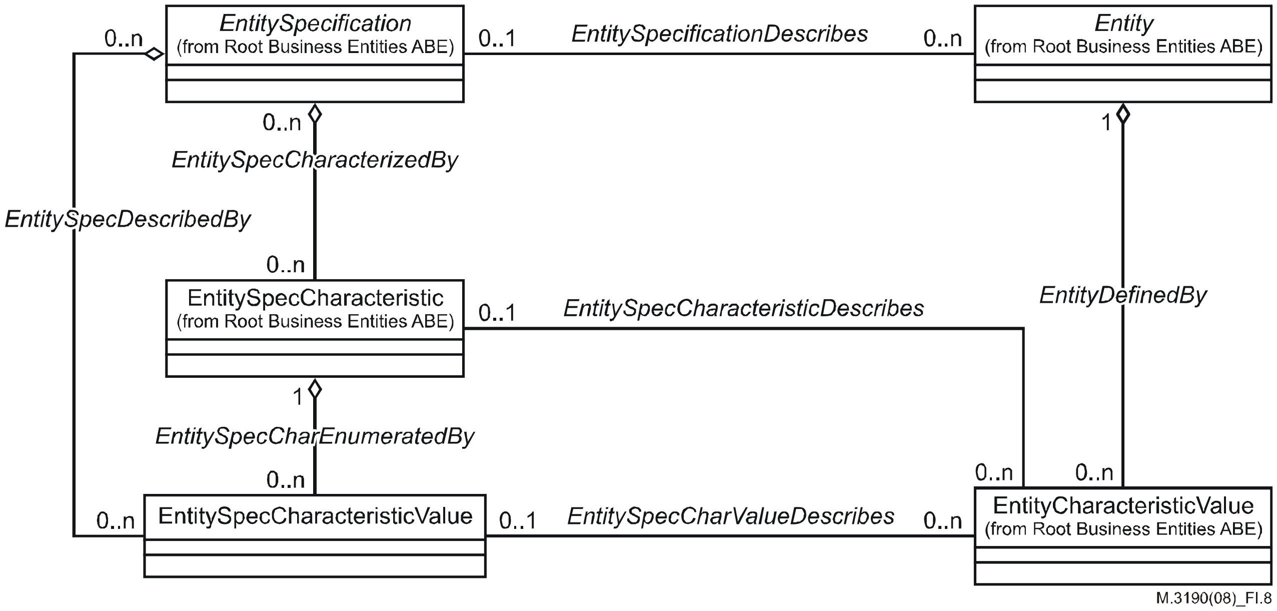
EntitySpecification
(from Root Business Entities ABE)
Entity
(from Root Business Entities ABE)
0..n
0..1
EntitySpecificationDescribes
0..n
0..n
EntitySpecCharacterizedBy
1
EntitySpecDescribedBy
0..n
EntitySpecCharacteristic
(from Root Business Entities ABE)
0..1
EntitySpecCharacteristicDescribes
EntityDefinedBy
1
EntitySpecCharEnumeratedBy
0..n
0..n
0..n
0..n
EntitySpecCharacteristicValue
0..1
EntitySpecCharValueDescribes
0..n
EntityCharacteristicValue
(from Root Business Entities ABE)
M.3190(08)_FI.8

Figure I.8 – Entity specification characteristic/entity characteristic pattern

In this pattern, instances of EntitySpecCharacteristic represent the various features/properties specific to an instance of an EntitySpecification. For example, one characteristic may be the title and start time for a video on demand (VoD) product offering. Instances of EntitySpecCharacteristicValue represent the values that features/properties can take on. For example, the titles of the VoD can be SuperMan I, SuperMan II, Mission Impossible I, and so forth.

Both EntitySpecCharacteristic and EntitySpecCharacteristicValues can be shared by multiple EntitySpecifications. That is the reason for the many-to-many association. For example, colour may be an option for many types of mobile devices. The association between EntitySpecCharacteristicValue and EntitySpecification exists because there may be a subset of all colours available for a particular instance of an EntitySpecification. For example, the entire set of colours may be red, green, blue, black, and silver, but for a particular instance of a mobile device, the colour options may be limited to red, black, and silver. Other features/properties, such as the start time for a VoD, are not constrained by a finite set of values, and therefore would not have associated instances of EntitySpecCharacteristicValue.

When an instance of an Entity described by an instance of an EntitySpecification is created, instances of EntityCharacteristic are created based on the EntitySpecCharacteristic. For example, when an instance of VoD is created a title selection is made, such as SuperMan II, and the start time is specified for the VoD, such as 12AM on the 21st.

Some features/properties are configurable, while others are not. For example, the title and start time are configuration for VoD. For other instances of EntitySpecification, a property, such as weight, may not be configurable. In this case, there may not be an instance of EntityCharacteristic as an entity.

Figures I.9 and I.10 show the use of this pattern within the product specification ABE. Figure I.9 shows the use of the entity specification component of the pattern. Figure I.10 shows the use of the entity component of the pattern.

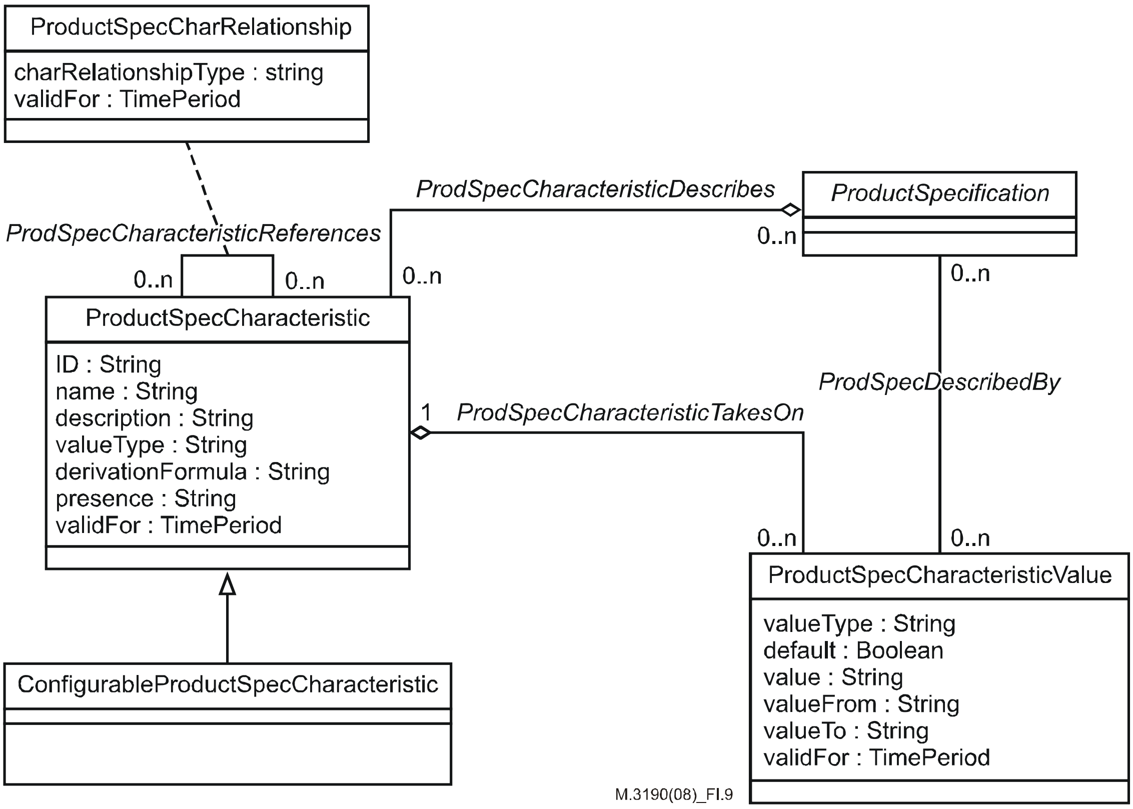

Figure I.9 – Product spec characteristic

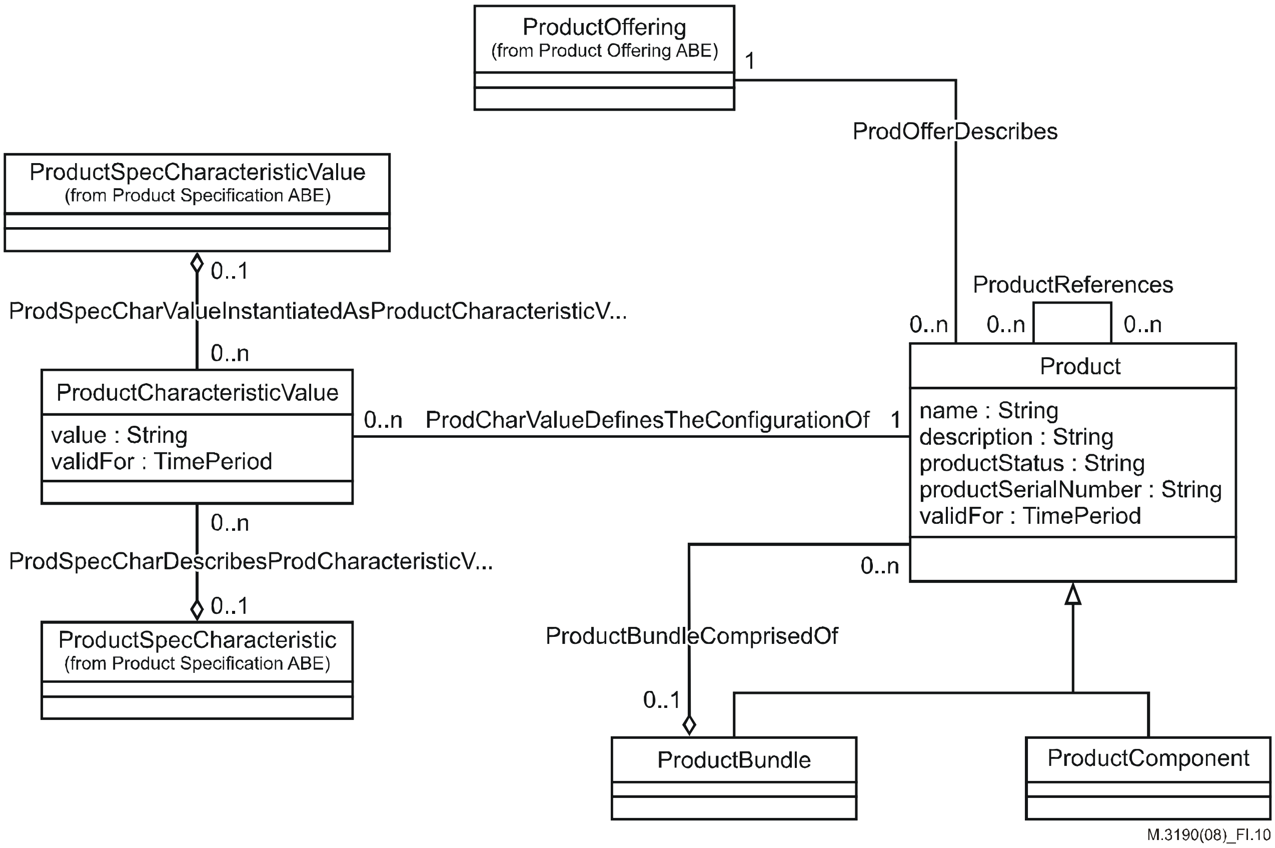
ProductOffering
(from Product Offering ABE)
1
ProdOfferDescribes
ProductSpecCharacteristicValue
(from Product Specification ABE)
0..1
ProdSpecCharValueInstantiatedAsProductCharacteristicV...
0..n
ProductReferences
0..n
0..n
0..n
Product
name : String
description : String
productStatus : String
productSerialNumber : String
validFor : TimePeriod
ProductCharacteristicValue
value : String
validFor : TimePeriod
0..n
ProdCharValueDefinesTheConfigurationOf
1
0..n
ProdSpecCharDescribesProdCharacteristicV...
0..1
0..n
ProductSpecCharacteristic
(from Product Specification ABE)
ProductBundleComprisedOf
0..1
ProductBundle
ProductComponent
M.3190(08)_FI.10

Figure I.10 – Product spec characteristic and product characteristic

Alternatives to this pattern do exist. For example, entities, such as hi-speed Internet, can be explicitly added to the model.

Appendix II

SID document structure

(This appendix does not form an integral part of this Recommendation)

The complete set of SID specifications is listed in Table II.1.

Table II.1 – SID document structure

Reference	Title	Included aggregate business entities
GB922 [TMF GB922]	**Business View Concepts, Principles, and Domains** describing the organizing structure of the SID business view shared information/data, the relationship to other TMF deliverables, and views of the contents of the model.	
GB922 Addendum 3 [TMF GB922A3]	**Product**	Product specification, product offering, product, product offering price
GB922 Addendum 4SO [TMF GB922A4SO]	**Service**	Service specification, service
GB922 Addendum 4-QoS	**Service quality of service**	Quality of service
GB922 Addendum 5PR	**Physical resource**	Physical resource
GB922 Addendum 5LR	**Logical resource**	Logical resource
GB922 Addendum 1P	**Party**	Party
GB922 Addendum 1L	**Location**	Location
GB922 Addendum 1BI	**Business interaction**	Business interaction
GB922 Addendum 1A	**Agreement (including service level agreement)**	Agreement
GB922 Addendum 1BT	**Base types**	Base types
GB922 Addendum 1T	**Calendar**	Calendar
GB922 Addendum 1J	**Project**	Project
GB922 Addendum 1POL	**Policy**	Policy
GB922 Addendum 1R	**Root business entities**	Root business entities
GB922 Addendum 2	**Customer**	Customer, customer order, customer SLA

SID can be applied in parts, i.e., there is no need to implement all domains in order to make use of SID. As an example, the Location domain is referenced by the Product and Service domains, but other means of supporting address requirements are also allowed.

Appendix III

SID use of UML

(This appendix does not form an integral part of this Recommendation)

III.1 UML overview

The unified modelling language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing object-oriented software and software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

III.2 Goals of UML

The primary goals in the design of the UML are:

- Provide users with a ready-to-use, expressive visual modelling language so they can develop and exchange meaningful models
- Provide extensibility and specialization mechanisms to extend the core concepts
- Be independent of particular programming languages and development processes
- Provide a formal basis for understanding the modelling language
- Encourage the growth of the OO tools market
- Support higher-level development concepts such as collaborations, frameworks, patterns and components
- Integrate best practices.

III.3 Why use UML?

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market. These techniques include component technology, visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale. In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The Unified modelling language (UML) was designed to respond to these needs.

III.4 UML diagrams

The underlying premise of UML is that no one diagram can capture the different elements of a system in its entirety. Hence, UML is made up of nine diagrams that can be used to model a system at different points of time in the software life cycle of a system. The nine UML diagrams are:

Use case diagram: The use case diagram is used to identify the primary elements and processes that form the system. The primary elements are termed as "actors", and the processes are called "use cases". The use case diagram shows which actors interact with each use case.

Class diagram: The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

Object diagram: The object diagram is a special kind of class diagram. An object is an instance of a class. This essentially means that an object represents the state of a class at a given point of time while the system is running. The object diagram captures the state of different classes in the system and their relationships or associations at a given point of time.

State diagram: A state diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object's state from an initial state to a final state in response to events affecting the system.

Activity diagram: The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

Sequence diagram: A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

Collaboration diagram: A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.

Component diagram: The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.

Deployment diagram: The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.

III.5 UML diagram classification – Static, dynamic, and implementation

A software system can be said to have two distinct characteristics: a structural, "static" part and a behavioural, "dynamic" part. In addition to these two characteristics, an additional characteristic that a software system possesses is related to implementation:

Static: The static characteristic of a system is essentially the structural aspect of the system. The static characteristics define what parts the system is made up of.

Dynamic: The behavioural features of a system; for example, the ways a system behaves in response to certain events or actions are the dynamic characteristics of a system.

Implementation: The implementation characteristic of a system is an entirely new feature that describes the different elements required for deploying a system.

The UML diagrams that fall under each of these categories are:

- *Static*
  - Use case diagram
  - Class diagram

- *Dynamic*
  - Object diagram
  - State diagram
  - Activity diagram
  - Sequence diagram
  - Collaboration diagram
- *Implementation*
  - Component diagram
  - Deployment diagram

Appendix IV

SID and the NGOSS views

(This appendix does not form an integral part of this Recommendation)

The NGOSS approach to developing SID in support of business applications can be characterized by the four NGOSS views, as shown in Figure IV.1. Each view exposes the aspects of the SID required to make decisions at various stages of the SID's development life cycle.

The information models of the SID are limited to the business view, but can be transformed into systems views and implementation views. The SID remains at the conceptual level.

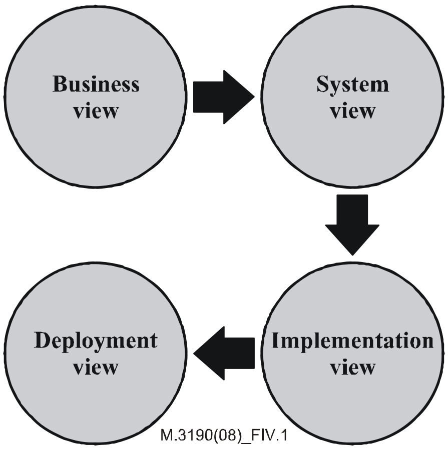

Figure IV.1 – NGOSS views

Development progresses from one view to the next. Each view has a different focus on a common set of SID artifacts that evolve from one view to the next.

The business view describes the conceptual, or logical, design of an information architecture from the business perspective. The system view adds behavioural aspects to the conceptual design beginning to define how the entities within an ABE function together as a unit. The implementation view describes the actual implementation of the information architecture in the form of a database definition or API interfaces. The deployment view describes the roll-out of the information architecture as a functioning database or functioning APIs.

IV.1 SID life cycle

The focus of the SID's development in the business view is on the discovery of business entities, their associations with other business entities, attributes that describe the entities, and types of entities.

For example, suppose there is a requirement to develop an application that supports an enterprise's Product Catalog. This is the first step in gathering information requirements. At this point, a Product Catalog may be an aggregate business entity (ABE) or a single entity. By defining Product Catalog, more aspects of its make-up can be ascertained. The definition of a Product Catalog is "A list of ProductOfferings for sale, with prices and illustrations, for example in book form or on the web. ProductCatalogs can be used by Customers during a self-care ordering process and may be used across one or more DistributionChannels". From this definition, an initial model for Product Catalog can be constructed as shown in Figure IV.2.

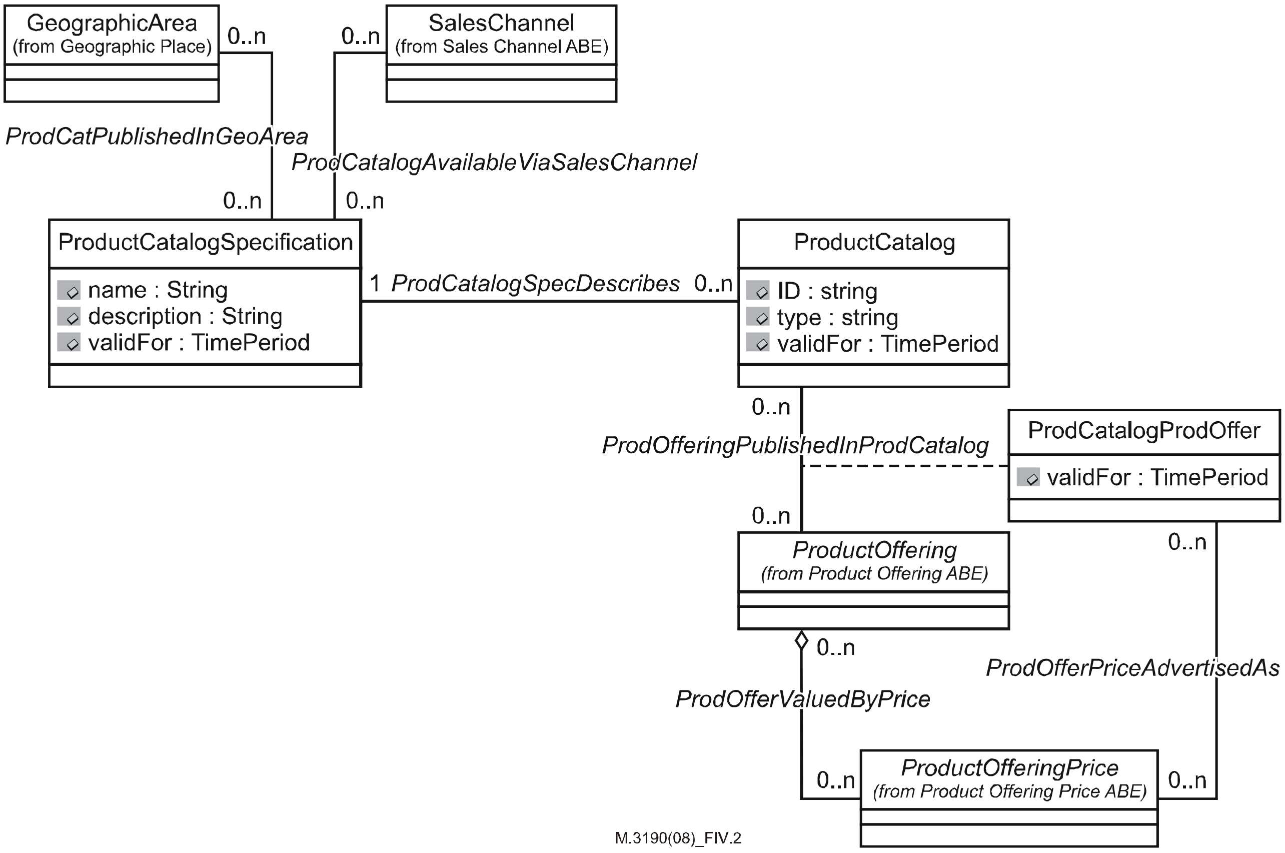

Figure IV.2 – Initial product catalog model (example)

Additional requirements gathered specified that the Product Catalog is particular to one or more geographical areas and to one or more sales channels. When building the model, the SID EntitySpecification/Entity pattern was employed resulting in a specification for the ProductCatalog as shown in Figure IV.2. Based on this model, it is assumed that ProductCatalog represents a cohesive collection of business entities. Therefore, it is classified as a second level ABE within the Product Offering ABE within the SID business view. This example also shows how an ABE can be identified using a bottom-up discovery process, as initially it was not known whether ProductCatalog was an entity or an ABE.

As development progresses, system view characteristics are discovered that further describe the Product Catalog ABE. Figure IV.3 shows several examples of system view characteristics, including the introduction of the ProdCatalogProdOffer association class and the operations on the ProductCatalog and ProdCatalogProdOffer entities.

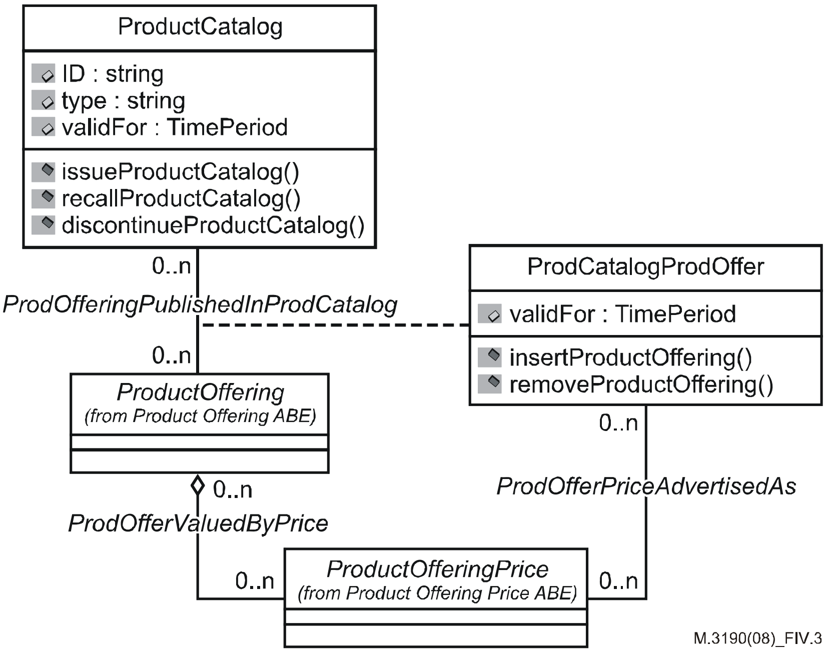

Figure IV.3 – Product catalog system view (example)

Once the system view is stabilized, definition of the implementation view commences. This view is the purview of other NGOSS teams, such as multi-technology operations system interfaces (MTOSI), and other TM Forum technical teams, such as the OSS through Java.

IV.2 SID logical and physical perspective

While the four NGOSS views are important in understanding the development of the SID model, the separation of the SID between a logical perspective and a physical perspective is crucial to implementing the SID. The reason is that multiple physical implementations of the SID can be derived either manually or automatically from a single logical model. This separation is shown in Figure IV.4.

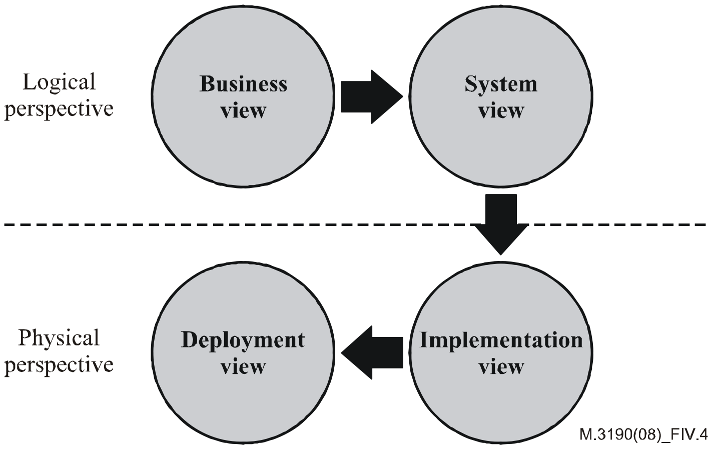

Figure IV.4 – SID logical and physical perspectives

At some point during the development of the SID logical perspective, implementation-specific modifications may or may not be made depending on how close a SID modeler wants to keep to a pure logical perspective. These changes may also be made during the initial development of the implementation view.

Appendix V

Positioning SID to the ITU-T M.1400-series Recommendations

(This appendix does not form an integral part of this Recommendation)

Since the initial SID submission includes the SID service and product addenda, it is considered necessary to "position" this submission relative to [b-ITU-T M.1402], *Formalization of data for service management*, in particular, and to the [b-ITU-T M.1400.x] series Recommendations in general. In providing such a positioning function, this appendix explains, at a high level, the relationship of the SID, as a conceptual information model, to the external terminology schemas of the [b-ITU-T M.1400.x] series Recommendations.

In terms of Appendix III of [ITU-T M.3020], *Comparison with Rec. ITU-T Z.601*, the requirements underlying a conceptual information model and the requirements underlying an external terminology schema fall into two classes: business requirements and specification requirements. SID, as a common information framework, is a member of the first class. As such, the SID model is organized into a taxonomy, or classification of business entities at various levels and associations between these. The scope of the SID business view is to cover a large portion of the information needs of a service provider that can be used as a reference for data that must be shared in implementing eTOM processes. According to [b-ITU-T Z.601], *Data Architecture of one software system*, the SID model would be an example of a concept schema to which multiple external terminology schemas could be mapped.

As a member of the specification class, the [b-ITU-T M.1400.x] series Recommendations focus on specification of the external terminology and grammar as perceived by the end users. The specification requirements may include requirements to support end-user interaction at their human-computer interfaces. Some of these requirements may specify syntactical requirements to be supported over any management interface. Syntactical requirements correspond to external terminology schemata of the data architecture as described in [b-ITU-T Z.601].

External terminology schemas define identifiers, name spaces, name bindings and other data that are to be efficient in use for the end users. This may lead to new object classes and relationships between object classes as compared to a pure conceptual approach. Hence, conceptual definitions of the information to be conveyed by management interfaces are not sufficient, as data syntax requirements may lead to additions and modifications of the conceptual definitions. (Methods for design of efficient end user data are found in [b-ITU-T Z.352], *Data oriented human-machine interface specification technique – Scope, approach and reference model*.)

Given this high level distinction between conceptual models and external terminology schemas, and their different usages, it is not considered to be essential that the current SID model be harmonized with the [b-ITU-T M.1400.x] series Recommendations. There is evidence that an attempt to informally map ITU-T M.1401 to the SID model revealed far more differences than similarities between the two, as one would expect of specifications developed in different environments for different purposes. A true harmonization effort would require that the SID model be revised to include new object classes and relationships between object classes as specified in the [b-ITU-T M.1400.x] series Recommendations and vice versa.

Bibliography

[b-ITU-T M.1400.x] Recommendation ITU-T M.1400.x-series, *Designations for interconnections among operators' networks*. <http://web.itu.int/rec/T-REC-M.1400/>

[b-ITU-T M.1402] Recommendation ITU-T M.1402 (2007), *Formalization of data for service management*. <http://web.itu.int/rec/T-REC-M.1402/>

[b-ITU-T Z.352] Recommendation ITU-T Z.352 (1993), *Data oriented human-machine interface specification technique – Scope, approach and reference model*. <http://web.itu.int/rec/T-REC-Z.352/>

[b-ITU-T Z.601] Recommendation ITU-T Z.601 (2007), *Data architecture of one software system*. <http://web.itu.int/rec/T-REC-Z.601/>

[b-Abstract] Woolf, Bobby, *The Abstract Class Pattern* <http://jerry.cs.uiuc.edu/~plop/plop97/Proceedings/woolf.pdf>

[b-ACIA] Alliance Common Information Architecture (AT&T, BT & Concert) version 3.1, 15 Dec. 2000.

[b-Distilled] Fowler, M., *UML Distilled: A Brief Guide to the Standard Object Modeling Language*.

[b-DDD] Evans, Eric, *Domain-Driven Design: Tackling Complexity in the Heart of Software*, ISBN: 0321125215 (Aug. 2003 release).

[b-DEN] Directory Enabled Networks (DEN). Strassner, J., *Directory Enabled Networks*, Macmillan Technical Publishing, ISBN 1-57870-140-6.

[b-Design] Stevens, WP, Myers, GJ & Constantine, LL (1974), *Structured Design*, IBM Systems Journal, Vol. 13, No. 2.

[b-FEA] US Government Federal Enterprise Architecture. <http://www.feapmo.gov/>

[b-IBM] IBM Research Design Patterns Web Pages. <http://www.research.ibm.com/designpatterns/publications.htm>

[b-ISO 19108] ISO 19108:2002, *Geographic information – Temporal schema*. <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26013>

[b-Java] Coad, Peter, Lefebvre, Eric, De Luca, Jeff (1999), *Java Modeling in Color with UML*, Prentice Hall PTR.

[b-Language] Christopher Alexander, *A Pattern Language: Towns, Buildings, Construction* Sara Ishikawa, Murray Silverstein.

[b-Patterns-1] Fowler, Martin, *Analysis Patterns – Reusable Object Models*, ISBN 0-201-89542-0 <http://martinfowler.com/articles.html>

[b-Patterns-2] Larman, Craig, *Applying UML and Patterns*, Second Edition ISBN 0-13-095004-1. <http://www.craiglarman.com/book_applying_2nd/Applying_2nd.htm>

[b-Patterns-3] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (Oct. 1994), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley, Reading, MA.

[b-Roles] Fowler, Martin, *Dealing with Roles*. <http://www.martinfowler.com/apsupp/roles.pdf>

[b-Rules] Moriarty, Terry, *The Grammar of Business Rules (3 parts)*. <http://www.tdan.com/>

[b-TMF GB922 primer] TM Forum GB922, *Shared Information/Data (SID) Model*. Addendum 0 – SID Primer, Version 1.1. <http://www.tmforum.org/DocumentsInformation/GB922InformationFramework/35040/article.html>

[b-UML] Booch, Rumbaugh & Jacobson, *The Unified Modeling Language User Guide*.

[b-Zachman] Zachman Framework http://www.zifa.com/framework.html. Various Articles by John A Zachman http://www.brcommunity.com/.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	**Telecommunication management, including TMN and network maintenance**
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2009