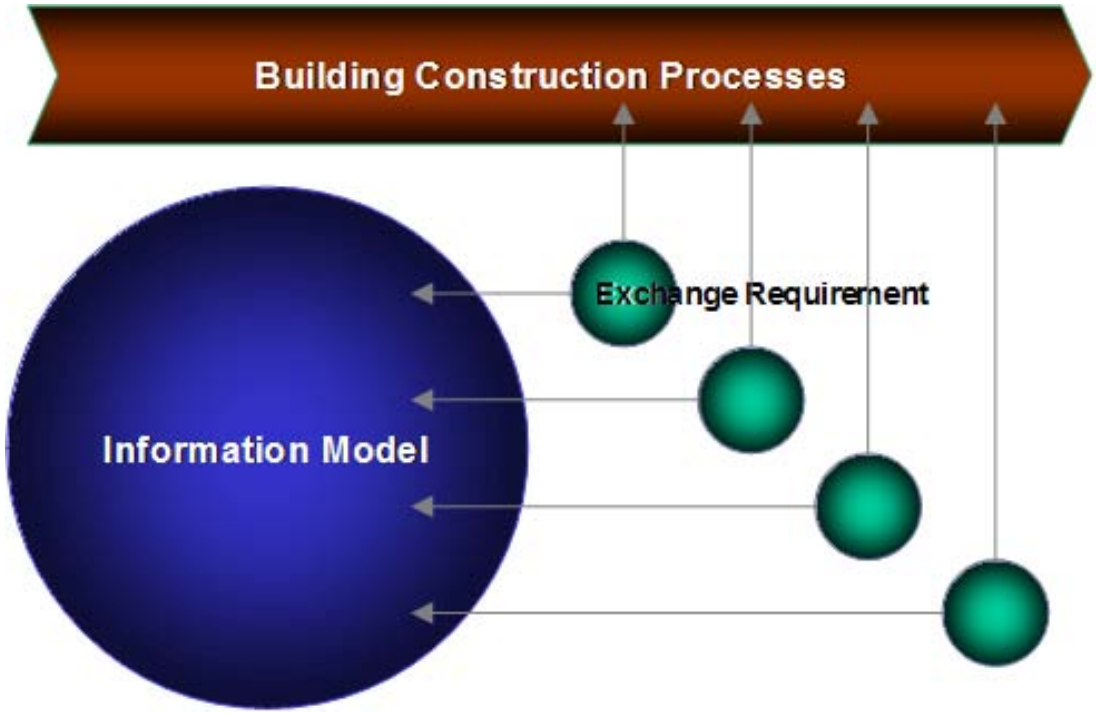
Document id	Title	Organisation /Author	Date	Status
IDMC 004	Information Delivery Manual	IUG/IDMC	05/12/10	Ver.1.2

Information Delivery Manual

Guide to Components and Development Methods



Revision history:

Date	Author	Version	Changes
28.02.2007	JDW	1.0 Draft	Issued
27.05.2007	JDW	1.0 Final	Issued
22.10.2007	JDW	1.1	Harmonization with Model View Definition. Extensions to Confluence editing. Organization section deleted. Exchange requirement model added and compile schemas section moved to body of report. Information Exchange Framework replaces IDM/MVD Harmonization and is moved forward in the body. Rules section modified and extended
05.12.2010	JK	1.2 Rev.	Changed document template. Added list of figures. Harmonized bullet lists, Changed fonts. Removed figures that only had the caption. Added space in tables.

Authors

- Jeff Wix, original edition
- Jan Karlshøj, revision

Table of Contents

Figures

1. Introduction

Building Information Modelling (BIM) is a new approach to describing and displaying the information required for the design, construction and operation of constructed facilities. It can bring together the numerous threads of different information used in construction into a single operating environment - reducing, and often eliminating, the need for the many types of paper document currently in use.

However to use BIM effectively and for benefit to be unlocked, the quality of communication between different participants in the construction process needs significant improvement. If the information required is available when it is needed and the quality of information is satisfactory, the construction process will itself be significantly improved.

For this to happen, there must be a common understanding of the building processes and of the information that is needed for and results from their execution.

The Industry Foundation Classes (IFC) provide a comprehensive reference to the totality of information within the lifecycle of a constructed facility. It has been created as an integrated whole in response to business needs identified by the international building construction community. It does not incorporate, however, a comprehensive reference to the individual processes within building construction.

The case for a comprehensive reference to processes in building construction is clear and compelling. By integrating information with process, the potential value of such a reference is greatly enhanced to become a key tool in driving the benefits of BIM.

The Information Delivery Manual (IDM) aims to provide the integrated reference for process and data required by BIM by identifying the discrete processes undertaken within building construction, the information required for their execution and the results of that activity. It will specify:

- where a process fits and why it is relevant
- who are the actors creating, consuming and benefitting from the information
- what is the information created and consumed
- how the information should be supported by software solutions

Doing this will make IFC more closely reflect real project needs and accelerate the use of BIM in real projects. It will also help to realize integrated project information as the key driver to the process improvement set as a goal for the AEC/FM industry by many industry commentators.

The IDM benefits both BIM users and software solution providers:

- For BIM users, it provides an easy to understand, plain language description of building construction processes, the requirements for information to be provided to enable processes to be carried out successfully and the expected end results of the processes. This will help to:
  - make information exchange between project participants more reliable
  - improve information quality
  - improve decision making
  - do a BIM project far more effectively

- For BIM software developers, it identifies and describes the detailed functional breakdown of processes and the IFC capabilities needing to be supported for each functional part in terms. This will help to:
  - respond better to user needs
  - guarantee quality of information exchange
  - provide for reusable software components

1.1 Scope

The scope of this document, section 1, is to present the reasons for development of the Information Delivery Manual (IDM), describe its components, define a methodology to be used in their

development and provide useful information about how to create and use its various parts. It provides a complete reference that step through the ideas and parts of the IDM.

IFC is an incredible expression of what the AEC/FM industry can do when it decides to work together to solve a problem. It succeeds really well at dealing with the whole field of information in AEC/FM projects across the lifecycle. The difficulty for software solution providers and industry users is that it is too unwieldy to use to handle the information needs of a particular process at a particular stage of a project.

Section 2 of the methodology describes the reasoning behind why the IDM was developed. In particular, it describes the need to break down the IFC schema into usable parts that have a more immediate value to users and solution providers so that they can build more quality and trust into their IFC based information exchange.

Section 3 introduces the component structure of the IDM and explains the way in which this structure meets the information exchange needs of users and solution providers. The key components of IDM are introduced and are explained in more detail in later sections.

Section 4 sets out the objectives of the IDM in terms of its vision, mission, goals whilst section 5 describes the groups of people at which it is targeted.

Section 6 describes an Information Exchange Framework that marries the Information Delivery Manual with the Model View Definition method used for documenting the certification of IFC compliant software by the International Alliance for Interoperability. It also describes how the framework extends IDM to providing guidance for users and software developers.

Section 7 describes the technical architecture that connects the IDM components together in a coherent structure that supports the information exchange needs of both users and solution providers. It then goes on to describe each of the components individually identifying their purpose and content, how they should be created and how used.

Section 8 provides guidance on development processes for IDM. These start from defining the scope for an IDM development, defining a project plan and acquiring development resources. It then goes on to look at the three primary development routes currently understood for IDM. These are conventional process modelling and data discovery (which can also be used for IFC extension), reverse engineering data requirements from software dialogues and extension of exchange requirements through specification of additional business rules. Post development issues are also considered in terms of terminology capture, publication of components using the IDM ‘Wiki’ publication method and the handling and management of issues.

Section 9 outlines some preliminary guidance on implementation of IDM components. It describes simple approaches to mapping IDM concepts to those that are internal to software applications

The IDM methodology also includes a number of appendices that are useful in guiding development and use of the IDM.

Appendix A describes how to define and localize actor specifications. Extensive use is made of the actor tables from the Omniclass classification system. This is an example of what is possible and not a recommendation for use.

Appendix B shows how the default project stages for IDM are defined and how they may be varied to reflect local usage and local naming. The HOAI process specification from Germany is referenced as an example.

Appendix C describes the naming rules used for various parts of IDM. These naming rules are an initial step in a more ambitious programme of IDM development that is intended in the long term to lead to contracted workflow provisions for building construction projects.

Appendix D describes particular methods applied by IDM development to the standard BPMN process modelling notation. This appendix should be used in conjunction with the general BPMN modelling guide published in conjunction with the IDM methodology or with any other useful BPMN modelling guidance.

Appendix E provides specific guidance on approaches used in the publication of IDM using the Confluence 'Wiki' programme. It does not provide instructions or guidance on using the Wiki itself since this is readily available within the software.

Appendix F provides guidance on creating IDM components within EXPRESS-G notation diagrams.

Appendix G provides information on generating and managing issues on IDM components using the JIRA issues management software applied generally within IFC development.

1.2 Terminology

The meanings of all terms used within the IDM Methodology are defined within the 'buildingSMART Glossary of Terms'.

1.3 Conventions Used

Within the Methodology, information has been grouped into different elements. Different elements are shown or marked in different ways to make them easier to identify. This 'convention' is used to bring particular points to the attention of the reader.

Descriptive text

The main body of this document is descriptive text that outlines the organization, creation and use of the IDM. Descriptive text is always shown in a plain font (Arial). This paragraph shows how descriptive text looks.

Examples

Examples are included within the Guide. Where the example includes text, this is shown in blue text using a serif font (Times New Roman). This paragraph shows how example text looks.

Code

```
The Guide also contains some information that looks like program code. This
is shown in a typewriter font (Courier). This paragraph shows how code text
looks.
```

2. Why an Information Delivery Manual

The Industry Foundation Classes (IFC) provide a comprehensive specification of the information about an AEC/FM project. It captures information from:

- all types of organization involved in the project (architects, engineers, constructors, facility managers etc.)
- all stages in the project lifecycle including initial requirements, design, construction, maintenance and operation.

IFC's have grown from requirements that are defined and tested by industry specialists. These requirements are developed in projects that focus on key areas of the AEC/FM process. On completion of a project, the results are integrated into the complete IFC Model. The final product of this integration is a single, consistent schema that aims to satisfy the needs of the AEC/FM industry.

Whilst this approach progressively satisfies the requirements for a complete schema, it does not take account of the value of the various developments that enable it.

For instance, most of the IFC development projects develop process maps describing the flows of information that are undertaken in practice (or that could be undertaken in an improved business process). However, on completion of the development, other than in the project document itself, no attempt is made to retain the process map nor to integrate the ideas of that process with any other process described in other development projects.

Similarly, the fully integrated IFC schema does not take account of the ways in which information is created and shared by practitioners. This is recognized within IFC development through the provision of views of the IFC schema. This is substantially enhanced through the 'Model View Definition effort' which improves the way in which views are specified. However, model views are targeted specifically at software developers and certification of their software and not at the way in which industry practitioners use the software.

As IFC compliant software solutions become a normal part of working on real projects it is important that secure, reliable and verifiable information exchange/sharing is enabled by software solution providers and applied by users. Users must be confident that the IFC data they are sending/receiving is accurate and sufficient for the activities they want to carry out. This is a need that is specific to the business processes undertaken within the industry.

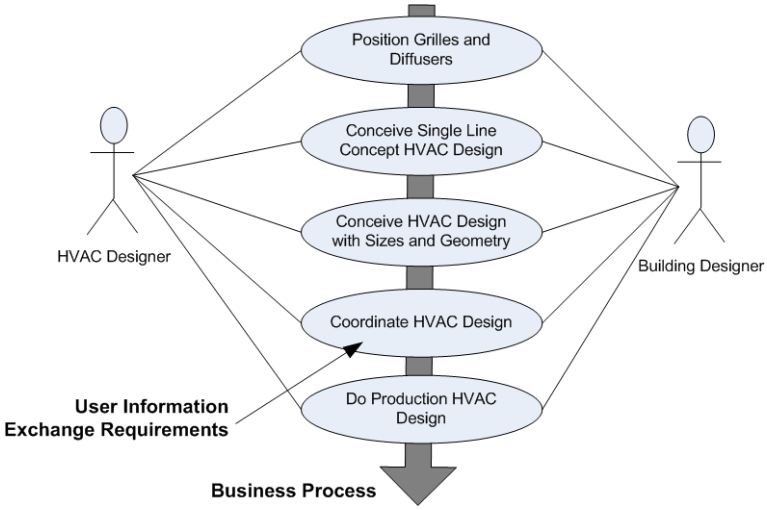


Figure 1: User requirements for information exchange

2.1 IDM as a Response

The Information Delivery Manual (IDM) is a response to these issues. It proposes a methodology that captures (and progressively integrates) business process whilst at the same time providing detailed user defined specifications of the information that needs to be exchanged at particular points within a project. To further support these user defined information exchange specifications, IDM also proposes a set of reusable modular functions that handle the basic information ideas in AEC/FM and that can be used to assist the development of further user defined information exchange specifications.

2.2 IFC Model Views

As the IFC model has matured and become stable, so the number of software solutions having IFC compliant interfaces has increased. IFC is now one of the most widely implemented international standard schemas.

A key issue for solution providers is what parts of the IFC schema they should implement. It is clear that most software solutions do not need to implement everything within the IFC schema to meet their needs. They can do so with a part of the schema. This is termed a 'view'.

A 'view' deals with the information requirements that need to be supported by a software solution in response to one or many business processes. For instance, the 'coordination view' meets information requirements for location and connection of building elements and for clash detection between them and serves the needs of multiple actor roles within the construction process.

IFC certification is carried out against views. Certification is a procedure for testing that an IFC compliant software solution will reliably and repeatedly export and/or import IFC data according to established testing criteria. Certification is important, as it is a primary means of providing the necessary trust that promotes IFC use on real AEC/FM projects.

Views are developed for certification using the IFC Model View Definition (MVD) approach. There is a close relationship between the development of the IDM information exchange requirements and IFC Model View Definitions and the Information Exchange Framework that harmonizes the IDM and MVD methodologies into a completely consistent approach recognizes this.

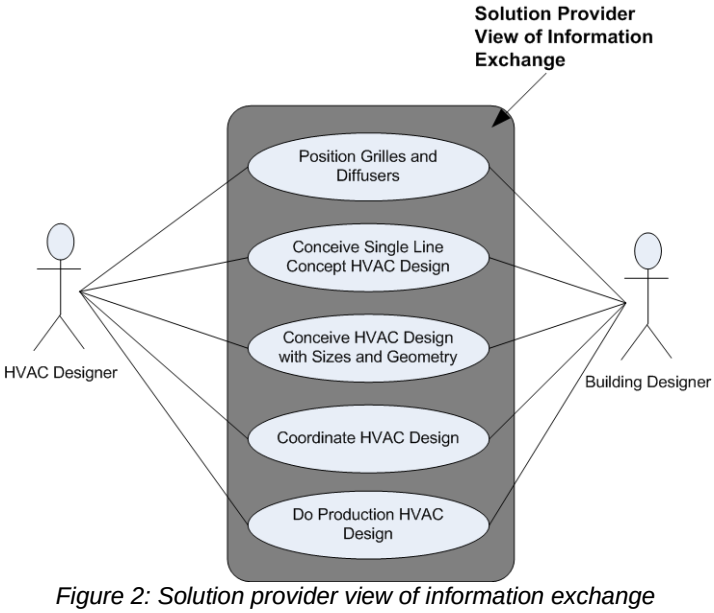


Figure 2: Solution provider view of information exchange

2.3 Using IDM for IFC Model Extension

IDM is specifically designed as a methodology that can break down the IFC schema into smaller, useful but still related parts. This would seem to contrast with projects that wish to extend the IFC schema into new areas that are not yet covered.

However, the approach is similar. The only difference between an IFC extension project and IDM development is that the entities to be specified in the information exchange requirements of an extension project may be proposed rather than actual. On the understanding that such exchange requirements would need to be revisited and corrected after IFC schema integration, it is therefore considered that the IDM methodology is also appropriate to IFC extension projects.

3. What is IDM

The complete IFC schema is developed as a set of individual topic schemas. Each topic schema typically represents a consistent overall idea (e.g. structural analysis, HVAC, cost, materials etc.). On completion, all of the topic schemas are brought together into the single schema which is the authorized working version**.** It contains hundreds of entities (classes), datatypes and property sets (the IFC components). This can be seen in fig. 3 below.

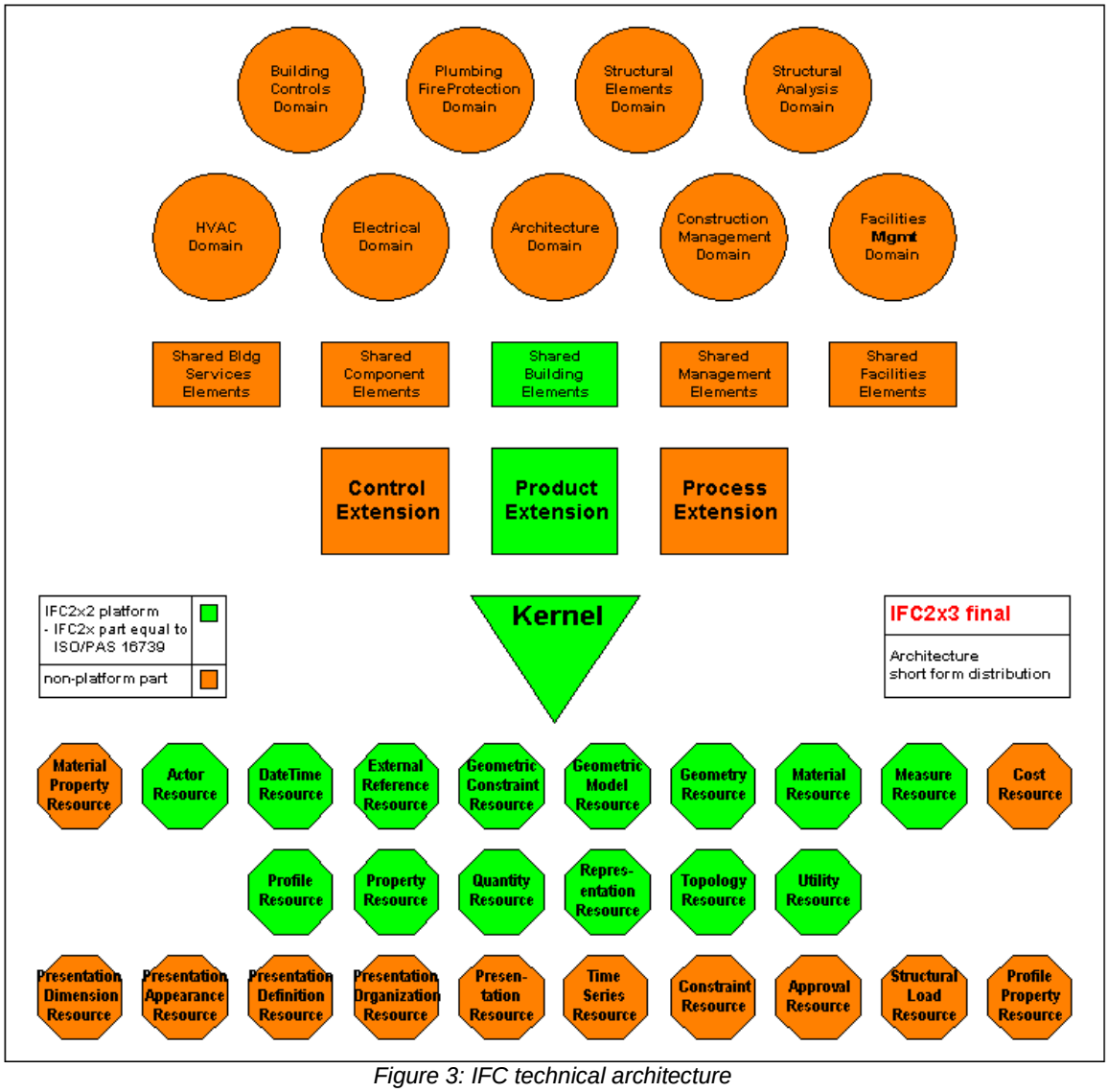
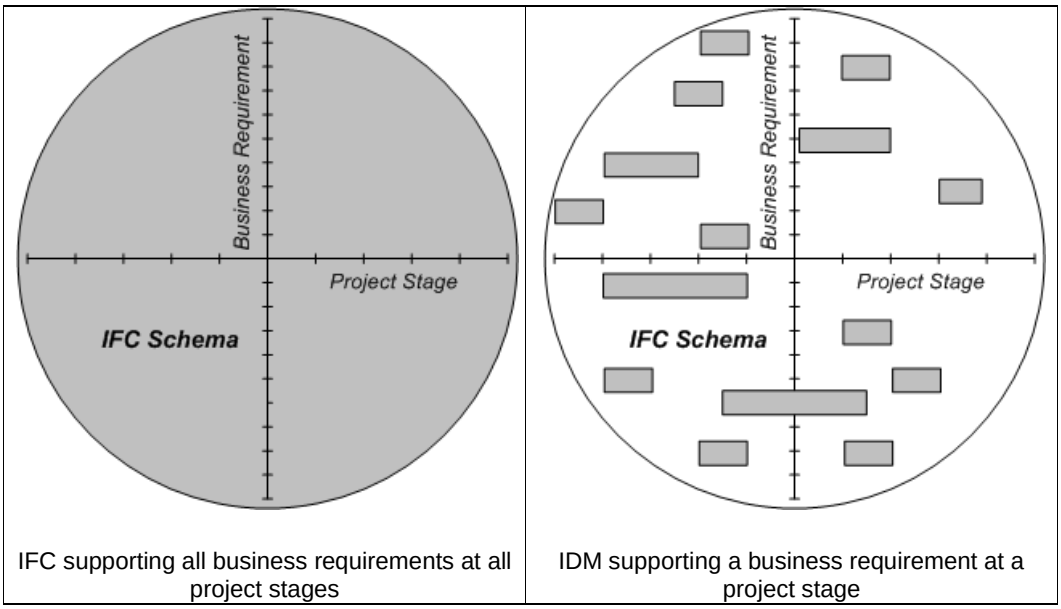


Figure 3: IFC technical architecture

This means that IFC is intended to support all business requirements at all project stages (see fig. 4). This is relevant where the whole of the project information needs to be shared or exchanged between organizations. But it is not the way that project information is usually delivered.



IFC supporting all business requirements at all project stages

IDM supporting a business requirement at a project stage

Figure 4: IDM support for business processes

It is more usual for information to be exchanged about a particular topic and the level of detail provided to be driven by the project stage. The need is (generally) to support one business requirement over one (or more) project stages (see Figure 4). This is a matter of deciding which IFC components should be used to meet requirements. This is important both for users and for solution providers. It is not just a case of selecting one of the development schemas since these are formed to help the task of distributing development between the model authors.

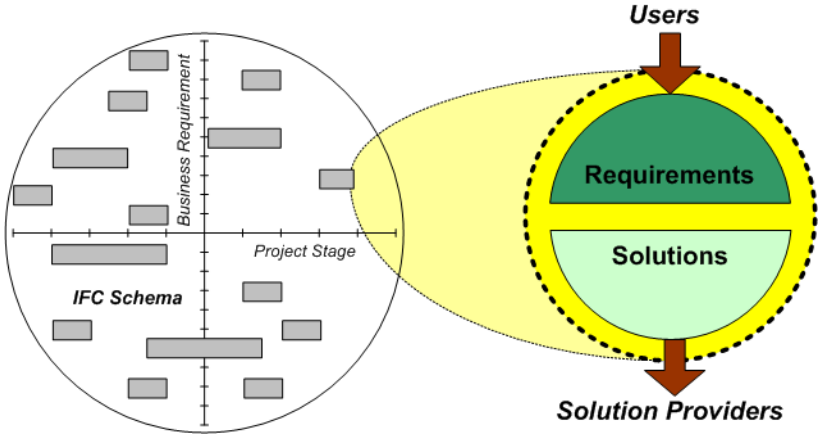


Figure 5: Business process requirements and solutions

Solution providers and software users means need to know which IFC components are important for their purpose:

- software users need to know so that they can be sure that IFC meets their needs,
- solution providers need to know to make sure they implement the right components AND meet the needs of their users.

3.1 Supporting the Business Requirement

To do this means that the set of information contained within IFC that needs to be exchanged to support a particular business requirement in the relevant project stages must be established. This is termed an information exchange requirement or, more simply, an '**exchange requirement**'.

An exchange requirement is intended to provide a description of the information in non technical terms. The principal audience for an exchange requirement is the user (architect, engineer,

constructor etc.) . It should however also be used by the solution provider since it provides the key to the technical detail that enables the solution to be provided.

3.2 Supporting the Software Solution

The technical content required by solution providers to support an exchange requirement is provided as a series of '**functional parts**' of the IFC schema. A functional part is a consistent and reusable set of units of information or 'concepts'

A functional part describes the information in terms of the IFC capabilities required. It also provides the schema for the information content as a subset of the complete IFC schema.

3.3 Supporting the Business Process

Many software solutions will actually support users in several exchange requirements that support an overall business process. The connection between exchange requirements and a business process is captured within a '**process map**'.

A process map typically deals with the development of information within the boundary of a particular topic. The complete set of IFC components that support a business process are identified within a 'Model View Definition'.

4. IDM Objectives

Having established why an Information Delivery Manual should be developed and what are the parts of it, the specific objectives for the IDM development can now be set down in terms of its vision, mission and goals.

4.1 Vision

The vision is that IDM will unite the description of business processes with the specification of information within the AEC/FM project lifecycle to enable realization of the full benefits of process improvement and information sharing.

4.2 Mission

The mission of the IDM is to provide:

- a comprehensive reference to information requirements for the AEC/FM industry by identifying
  - the processes that require the exchange or sharing of information between project participants,
  - the information required for and resulting from the execution of these processes.
- a basis for the consistent development of project specific process models
- a common basis for agreement between project participants about the information that can reasonably be expected to support a project specific process

4.3 Goals

The goals of the IDM are to:

- define the processes within the AEC/FM project lifecycle for which users require information exchange
- specify the IFC capabilities required to support these processes.
- describe the results of process execution that can be used in subsequent processes
- identify the actors sending and receiving information within the process by role
- ensure that definitions, specifications and descriptions are provided in a form that is useful and easily understood by the target group

5. IDM Target Groups

Three target groups for which IDM is designed include:

Executive User

An executive user is a person who:

- makes the decision to use IFC based information exchange
- is aware of the business process concerned and the impact that improvements in its execution will have at the business level
- does not need to have technical detail about the use of information within the process
- does not need to know about software development or the IFC model

End user

An end user is a person who:

- uses IFC for information exchange in practice
- needs to know exactly what information to expect and how to use it in the business process
- does not need to know about software development or the IFC model

Solution provider

A solution provider is a person or organization that:

- writes a software application with an IFC interface
- needs to know what executive and end users expect from their software solution
- needs to have a detailed technical knowledge of the IFC model

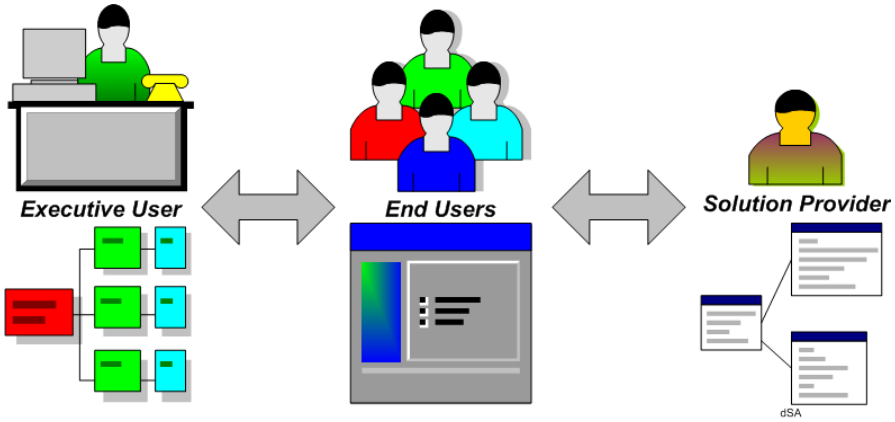


Figure 6: IDM target groups

6. Information Exchange Framework

IDM forms the first part of a complete framework that meets the information definition and exchange requirements of both users and software solution providers. This also includes a technology element that allows for configuration according to purpose, time and location. The second part of the framework is termed 'Model View Definition' and was approved as the means by which software certification is documented by the buildingSMART® in April 2006 and revised in 2010. The third part of the framework is termed 'IDM Deployment' and is about providing guidance to users about information preparation and software use for high quality information exchange.

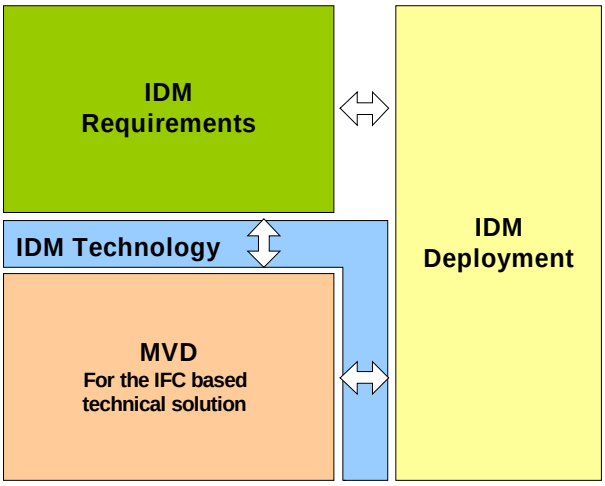


Figure 7: Overall architecture of the Information Exchange Framework

The framework starts with an information model that has been developed to meet the integrated set of information needs within a particular industrial context. For the AECO (or building construction and facilities management) industry, this is the IFC model.

From the IFC model, view definitions can be derived for particular purposes. The model view definition derived in this way can be seen as covering a well understood range of needs within a software application and consequently can be used to support the certification of such software, That is:

A model view definition (MVD) is the set of information from the information model that can be supported by a type of software application[1].

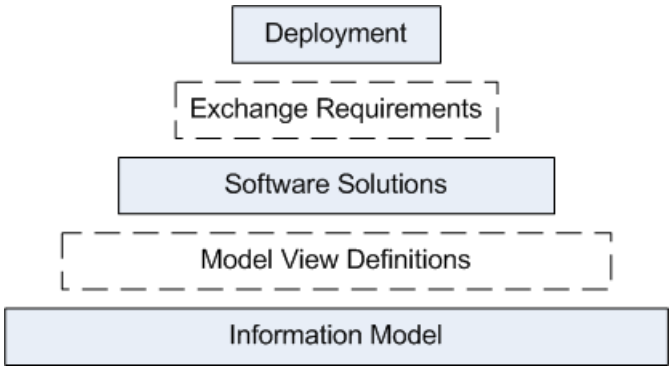


Figure 8: Layers of the Information Exchange Framework

1 A particular software application may conform to many types and may therefore support several model view definitions

From the IFC model, the subset of information can also be derived that supports the information exchange requirements of a user for a particular purpose, at a particular time within a project and at a particular location. That is:

An information exchange requirement (ER) is the set of information from the information model that is applicable for purpose, time and location

Since it is reasonable to expect that a software application should support a user for this purpose, time, and location then it can also be seen that the subset of information within this exchange requirement could also be derived from the model view definition subset. Similarly, if several subsets of information required to support the needs of the user at different times within a project are derived and each of these is supported by a particular software application (or type of software application), then it can be seen that the model view definition subset of information can be integrated (compiled) from the several exchange requirement subsets.

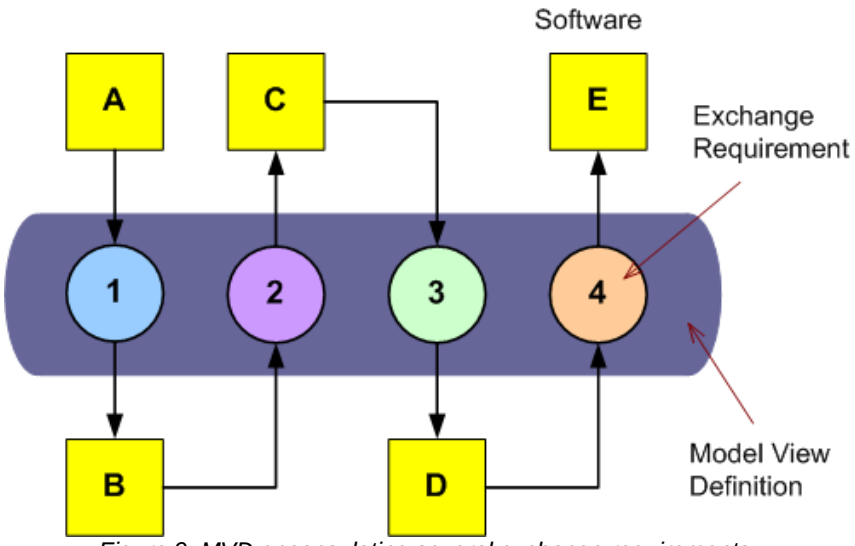


Figure 9: MVD encapsulating several exchange requirements

Refer to http://buildingsmart.be.no:8080/buildingsmart.com/standards/mvd for more complete details for the second part of the Information Exchange Framework i.e. that Model View Definitions.

7. IDM Components

Within this section, the various components of IDM are described, together with how they are developed. Initially, the relationship between components is described in a fundamental IDM technical architecture.

7.1 IDM Technical Architecture

The technical architecture of the IDM provides a generalized view of the various component parts and how they relate to each other.

7.1.1 Primary IDM Components

The IDM comprises several components each of which can be described as a layer within the architecture. The organisation of the components is based on two related ideas:

1. The components relate to process definitions and descriptions at the top layer, progress through data specifications in the middle layers and include application software elements at the bottom layer.
2. Similarly, the components relate to industry practitioners at the top levels of the architecture and to ICT analysts and programmers at the bottom levels.

The purpose of each component within the technical architecture is as follows:

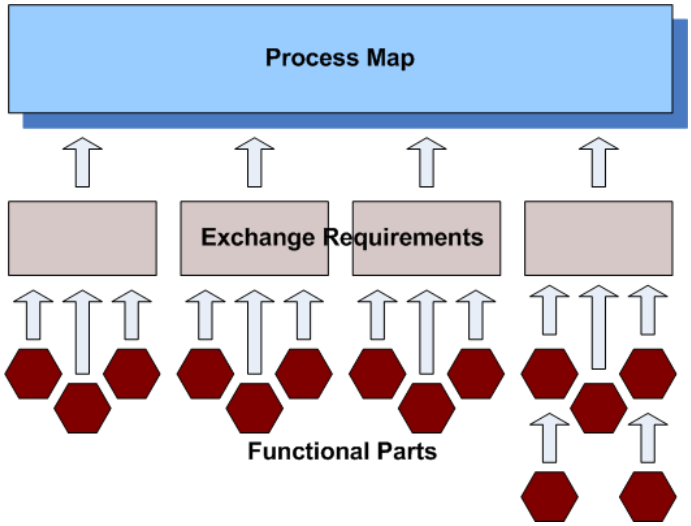


Figure 10: Basic IDM technical architecture

7.1.1.1 Process Maps

A process map describes the flow of activities within the boundary of a particular topic.

The purpose of a process map is to gain an understanding of the configuration of activities that make it work, the actors involved, the information required, consumed and produced.

7.1.1.2 Exchange Requirements

An exchange requirement is a set of information that needs to be exchanged to support a particular business requirement at a particular stage of a project.

Typically, for IDM as presently established, the set of information should be defined within the IFC model. However, the IDM approach will also work with sets of information defined within other industry

standard models such as the Geographic Markup Language (GML) as defined by the Open Geospatial Consortium (OGC).

An exchange requirement is intended to provide a description of the information in non technical terms. The principal audience for an exchange requirement is the user (architect, engineer, constructor etc.). It should however also be used by the solution provider since it provides the key to the technical detail that enables the solution to be provided.

7.1.1.3 Functional Parts

A functional part is a unit of information, or a single information idea, used by solution providers to support an exchange requirement.

A functional part describes the information in terms of the required capabilities of the industry standard information model upon which it is based. For IDM as presently established, the functional parts are based on versions of the IFC model.

A functional part is fully described as an information model in its own right as well as being a subset of the information model on which it is based.

7.1.1.4 Concepts

A concept is a fragment of information that can be used in a functional part (where it is bound to a release of the IFC model) or to an exchange requirement (where it is expressed in generic terms). It can be used to capture the basic functionalities within a model such as naming, identification etc. A concept does not need to be simply related to a single entity or even to a whole entity. For instance, the concept of a software identifier simply describes how a globally unique identifier attribute is asserted for an entity.

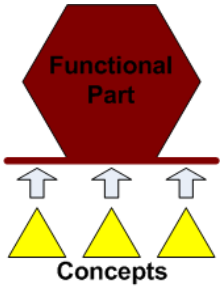


Figure 11: Concepts used with a functional part

Concepts are a shared level between exchange requirement models and model view definitions.

7.1.2 Implementing and Validating IDM Components

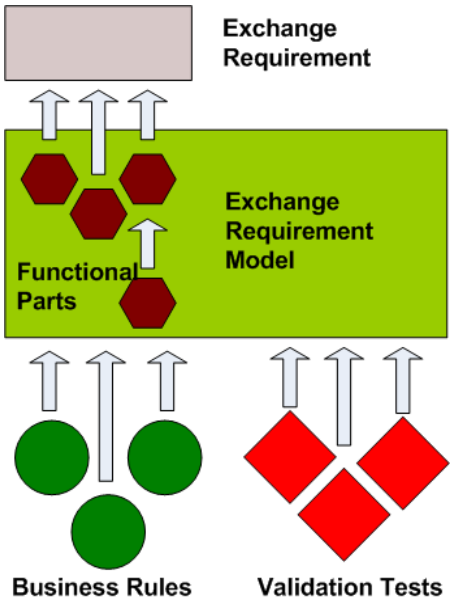


Figure 12: Defining an 'Exchange Requirement Model'

7.1.2.1 Exchange Requirement Model

An exchange requirement model is the technical solution of an exchange requirement. It is created from the set of functional parts defining the units of information that support the underlying exchange requirement.

Note that there is a 1:1 correlation between an exchange requirement model and an exchange requirement. Whilst the expression of an exchange requirement is completely independent of any schema or any particular version of a schema, an exchange requirement model that relates to it is schema dependent. This is due to the fact that it is created from schema dependent functional parts.

Exchange requirement models are particularly significant as they are the IDM components that:

- will be supported within software applications;
- form part of the model view definition that is certified;
- are the components to which business rules are applied;
- are the components against which validation tests can be applied.

For further information on how an exchange requirement model is created from functional parts, see Appendix F: Compiling IDM Express Parts.

7.1.2.2 Business Rules

Business rules describe operations, definitions and constraints that may be applied to a set of data used within a particular process or activity. They enable controls to be applied to:

- use of specific entities,
- attributes and properties that must be asserted (or not asserted),
- values, ranges of values or value limits that should be observed.
- dependencies between entities or attributes or attribute values.

Business rules can be used to vary the result of using an information model without having to change the information model itself. This provides the model with agility so that, through the application of different sets of business rules to the same information model, different local applications of the model can be defined.

Note that it is possible to add to, amend or even delete business rules without affecting the underlying information model.

Business rules may be expressed as formal propositions in terms of their actions on exchange requirements. However, they must be expressed in an appropriate coded form for specific actions on the functional parts that are contained within an exchange requirement.

An example of a business rule expressed as a proposition is the requirement that 'the area of a space whose type is "Executive Office" must be greater than or equal to 10m2'. In this form, it is applicable to the exchange requirement. When applied to a functional part, this is coded in the logical form appropriate to the manner in which the attribute/property is expressed.

7.1.2.3 Validation Tests

Validation tests are tests carried out on the information exported from a software application according to the schema of an exchange requirement model. They are used to ensure that a stated exchange requirement is being satisfied according to a set of applied business rules.

Validation tests must be carried out using test files that have a known performance and that are specifically designed to validate particular aspects of the exchange requirement model.

The values assigned to attributes and properties within a test file may vary between locations in which validation tests are carried out. This is because different sets of business rules may be applied to the same exchange requirement model in different places.

Validation tests are applied for the purposes of:

- verifying that the export of information from a software application meets the quality criteria set out in an exchange requirement
- improving the quality of software implementations
- providing metrics against which claims made for software performance can be verified
- making comparisons between software applications fulfilling the same objectives (when compared using the same tests)
- estimating reliability

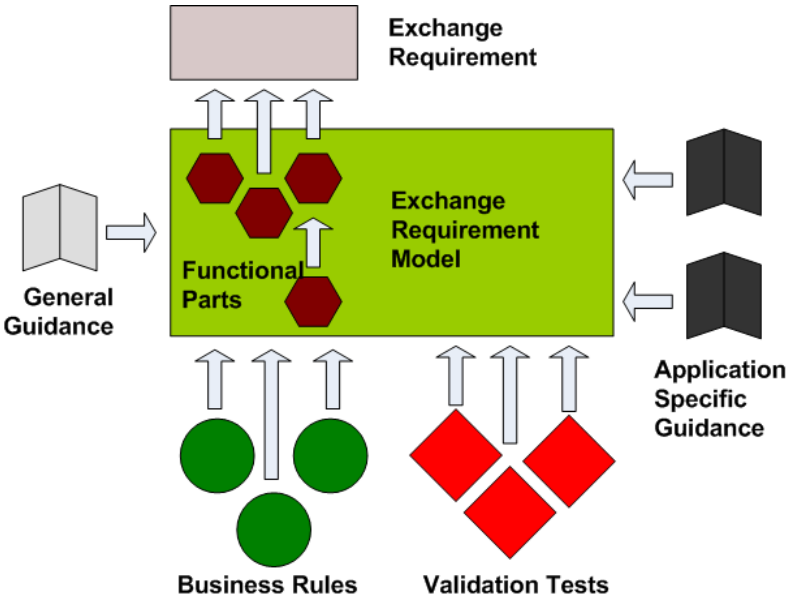


Figure 13: Providing general and application specific guidance

7.1.3 General and Application Guidance

7.1.3.1 General Guidance

General guidance is guidance provided to users about the extent and quality of the information that they will need to prepare within a building information model for export according to the provisions of an exchange requirement. It provides a preliminary statement of quality which, in conjunction with the application specific guidance, can be used to define the quality objectives against which the performance of building information model users can be tested.

7.1.3.2 Application Specific Guidance

Application specific guidance is particular guidance provided by solution providers or by other third party information providers that describes how a software application meets the business needs expressed by an exchange requirement. This may also include guidance on how to use the software in the circumstance of the exchange requirement and may also describe how the results are presented and applied.

Note that application software guidance is not provided as part of the IDM but is included within the technical architecture as an important and related provision.

7.1.4 Reference Processes

A reference process is an identifiable basic unit of a process map (or an activity) that can be considered to have a universally consistent definition both in terms of its meaning and its attributes/properties.

A reference process exists as a process type. A reference process may have many process occurrences within a building construction project.

The purpose of capturing a reference process is to support the progressive definition of a reference process library from which future industry standard and locally specific (including project specific) process maps can be developed. In the case of project specific process maps, it is considered that a set of reference processes can form a process ontology that can be used, with the addition of planning durations or schedule times, in the creation of a project plan and/or schedule

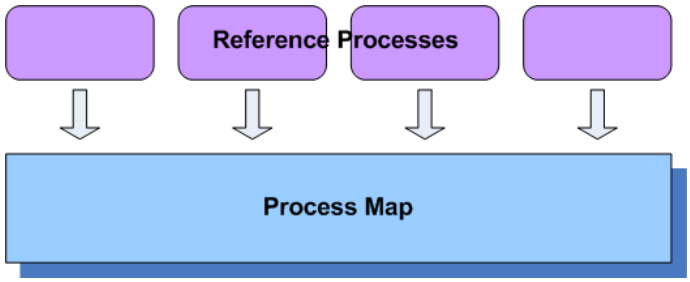


Figure 14: Reference processes in the IDM technical architecture

7.2 Process Map

The purpose of a process map is to help in understanding how work is undertaken in achieving a well defined objective. A process map:

1. has a Goal
2. has specific inputs (typically from other exchange requirements and from other data sources)
3. has specific outputs (typically to other exchange requirements)
4. uses resources
5. has a number of activities that are performed in some order
6. may affect more than one organizational unit
7. creates value of some kind for the customer

For the IDM, the principal roles of the process map are to:

- Set the boundary for the extent of the information contained within the process
- Establish the activities within the process and their logical sequence
- Identify the exchange requirements that support the activities within the process
- Enable reference processes to be determined.

The actual information that is within the process boundary is determined by the contents of the exchange requirements that support the activities within the process.

7.2.1 Notation

The preferred approach to developing a process map within IDM is to use the Business Process Modelling Notation (BPMN). The reasons for adopting this notation are:

- It is supported as an emerging standard by the Object Management Group (OMG)
- It is increasingly being used in the specification of business processes within major projects
- There are several available software tools that range from fairly simple, free applications that work with common industry solutions such as Visio to extensive industrial strength solutions.
- The notation has a conversion method to the Business Process Execution Language for Web Services (BPEL4WS) which is emerging as a standard XML based approach for workflow control.

Guidance on developing a general process map in BPMN is given in the related IDM document ‘Quick Guide: Business Process Modeling Notation (BPMN)’

IDM uses some of the techniques of BPMN in particular ways. Guidance on these techniques is given in Appendix D.

7.2.2 Content

All processes described within a process map should be related to the defined project stages as they appear in exchange requirements documentation.

A process map contains the following sections:

- header section
- overview section
- specification of processes
- specification of data objects
- specification of exchange requirement
- specification of coordination point gateways

7.2.2.1 Header Section

A header section provides administrative information about the process map and also identifies the exchange requirements that satisfy the requirements of the process. This includes:

Name	The name or title given to the process map. This should conform to the IDM naming rules
Identifier	A unique identifier that is assigned to a process map.
Change Log	Used to mark the creation of the process map and changes made to it over time. The date of creation/change should be included together with an identifier for the person responsible and a brief description of the changes made.
Exchange Requirements	Identifies by name the exchange requirements that are within the boundary of the process map (and that are specified further below)

An example of a completed header section is shown below.

Name	Cost Modelling

Identifier	NBS_PM_0001

Change Log		
2005-09-25	Created	jeffrey.wix@aec3.com

Exchange Requirements	er_exchange_cost_model (order_of_magnitude)
	er_exchange_cost_model (preliminary_appraisal)
	er_exchange_cost_model (approximate_estimate)
	er_exchange_cost_model (preconstruction_estimate)
	er_exchange_cost_model (request_for_quotation)
	er_exchange_cost_model (quotation)
	er_exchange_cost_model (claim_estimate)
	er_exchange_cost_model (variation_estimate)
	er_exchange_cost_model (final_account)

Example 1: Header section in process map

7.2.2.2 Overview

The overview section provides a comprehensive discussion of the overall process within the process map. Illustrations may be used to illustrate particular points within the overview.

The first part of the overview is used as an 'excerpt'. This is an abbreviated description of the process that is presented in the IDM web site. It should be used to provide a statement of the scope of the process map.

The remainder of the overview should extend the discussion and make clear the intended content and purpose of the process map.

7.2.2.3 Specification of Processes

The various diagrams created for the process map should be included. All processes within the maps should have an identifier and name.

Each process (or activity[2]) within the process map is described in such detail as required. The aim of the description is to be able to describe the intent of the process to the user. Providing that it achieves this objective, the extent of the description may be as long or as short as the creator requires.

Specify Usage Profile Data [ID:2.5.3]

Type	Task
Name	Specify Usage Profile Data
Documentation	Usage profiles are concerned with the time based usage of spaces and provide information about miscellaneous heat gains/losses to the space over time.These include: - Patterns of people within a space and their activities within the space - Patterns of machinery and equipment usage within the space - Patterns of artificial light usage within the space Typically, usage profiles are acquired with space type data. In this process, usage profiles can be added, adapted and amended for occurrence purposes.

Example 2: Process specification in process map

7.2.2.4 Specification of Data Objects

A data object is a named collection of data that is exported from or imported to a process map. For each data object that is not an exchange requirement, a short indication of its name and purpose should be provided.

Regulations

Type	Data Object
Name	Regulations
Documentation	Building regulations applicable to energy demand and comfort that must be satisfied. Analysis enables testing of results against these values.

Example 3: Data object specification in process map

7.2.2.5 Specification of Exchange Requirements

An exchange requirement is a particular type of data object within a process map that is located within the 'Information Model' actor role.

The description provided for an exchange requirement should be more complete than that given for a general data object. Typically, it is anticipated that the process map description for an exchange requirement should be equivalent to, and preferably the same as, the overview description within the exchange requirement documentation and should include an overall identification of the data within the process.

er_exchange_space_model [sketch]

Type	Data Object
Name	er_exchange_space_model [sketch]
Documentation	Building information model resulting from initial programming that identifies expected spaces.including expected shape and size. All spaces within the building must be available within the building model

2 The terms process and activity are used here as synonyms. In comparison to UML activity diagrams, a collapsed process is equivalent to an action that can be broken down into activities whilst a complete process or task is equivalent to an activity that cannot be further broken down.

	including non-functional spaces such as technical spaces, circulation spaces, shafts etc. It is not sufficient to define functional spaces such as offices and leave corridors as unidentifiable voids surrounded by geometry. Space information required includes: •guid as a non changing space identifier which remains with it for life •shape representation; must be at least a footprint and extrusion. If a 2D model is provided, them this should be the footprint of the space (in which case, a specification of space height would also be required) *note that space extrusions should be in + Z direction to relate to space boundaries (which must be extruded in + Z). Extruding in -Z causes problems.* •space type to which spaces correspond; this should include type name (according to an agreed local convention),. *note that space types may be designated differently for different functional purposes. e.g architecturally as 'type 1', HVAC as 'type 2' and electrically as 'type 3'* •name (specifically identifies architecturally defined room number or code e.g. 2/262). … etc.

Example 4: Exchange requirement specification in process map

7.2.2.6 Specification of Coordination Point Gateways

A coordination point gateway is a named point within a process map at which the information from exchange requirements is brought together to enable coordinated decision making to occur. For each coordination point gateway, a short indication of its name and purpose should be provided.

Coordinate_spaces_and_systems

Type	Coordination Point
Name	Coordinate_spaces_and_systems
Documentation	The purpose here is to bring together information that is currently available about the spaces as required for energy analysis and the technical systems designs. During the initial stages, information about systems may not be available. In this case, the energy analysis is speculative.

Example 5: Coordination point gateway specification in process map

7.3 Exchange Requirement

An exchange requirement represents the connection between process and data. It applies the relevant information defined within an information model to fulfil the requirements of an information exchange between two business processes at a particular stage of the project.

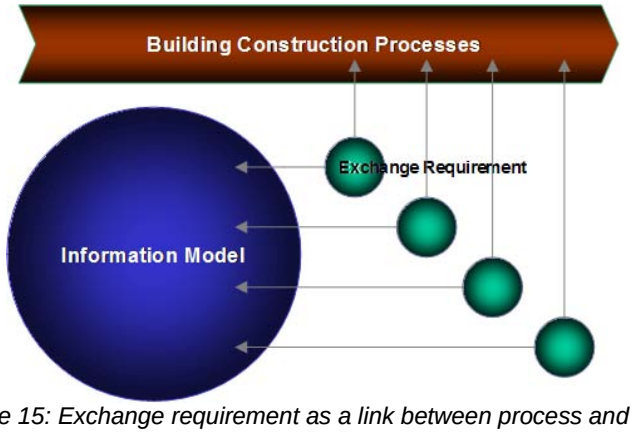


Figure 15: Exchange requirement as a link between process and data

An exchange requirement might be simple as in the case of an order that results from a purchasing process enabling a supplier to provide the required components. Alternatively, it might be complex as in the case of an architect providing a basic building model to an HVAC consultant to enable thermal analysis calculations to be undertaken.

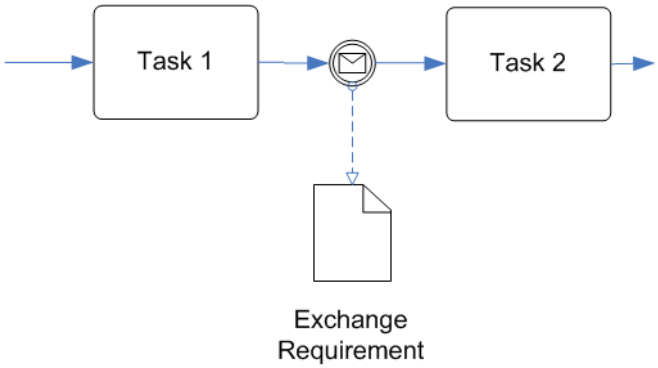


Figure 16: Exchange requirement for one downstream task

An exchange requirement describes a set of information from a process that has been performed by an actor to enable a downstream process to be performed by another actor. It is shown as the target of a message from a 'message driven event' in the IDM extensions to the BPMN notation.

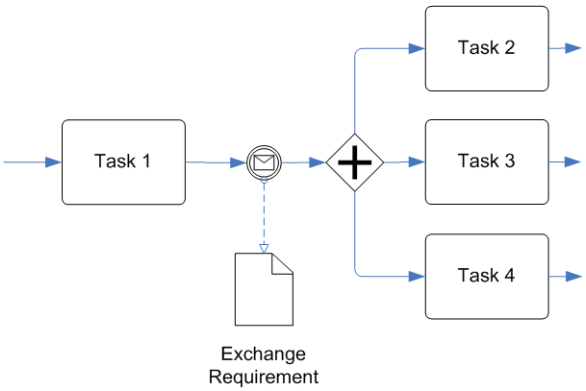


Figure 17: Exchange requirement for multiple downstream tasks

An exchange requirement may provide the required information for multiple downstream operations. This is shown using an 'AND' gateway in the above process model. For instance, a space model provided by a building designer may be used in energy analysis, HVAC design, and structural design.

7.3.1 Content

An exchange requirement contains the following sections:

- header section
- overview section
- information section

7.3.1.1 Header Section

A header section provides administrative information about the exchange requirement. This includes:

Name	The name or title given to the exchange requirement. This should conform to the IDM naming rules
Identifier	A unique identifier that is assigned to an exchange requirement.

Change Log — Used to mark the creation of the exchange requirement and changes made to it over time. The date of creation/change should be included together with an identifier for the person responsible and a brief description of the changes made.

Project Stage — Identifies the project stage (s) for which the exchange requirement is used. An exchange requirement may be applicable to one or several project stages. Careful thought should be given to the assignment of an exchange requirement to project stages. This information may be significant to solution providers and will be important to future developments of process standards using exchange requirements.

Project stages should be consistently applied to all exchange requirements. IDM uses a list of stages based on the development of the Generic Process Protocol

An example of a completed header section is shown below.

Name	Exchange HVAC Model (Equipment)

Identifier	NBS_ER_0001

Change Log		
2006-03-07	Created	jeffrey.wix@aec3.com

Project Stage	0	Portfolio requirements	
	1	Conception of need	
	2	Outline feasibility	
	3	Substantive feasibility	
	4	Outline conceptual design	✓
	5	Full conceptual design	
	6	Coordinated design and procurement	
	7	Production information	
	8	Construction	
	9	Operation and maintenance	
	10	Disposal	

Example 6: Header section in exchange requirement

7.3.1.1 Overview

An overview states the aims and content of the exchange requirement in terms that are familiar to the user. The aim is that it should be understood by a person who needs to be aware of what the exchange requirement is intended to achieve but who does not need to know the detail of how it is achieved. Typically, this person will be an ‘executive user’; an example would be someone acting in a project management role.

Illustrations may be used to illustrate particular points within the overview.

The first part of the overview is used as an ‘excerpt’. This is an abbreviated description of the exchange requirement that is presented in the IDM web site. It should be used to provide a statement of the scope of the exchange requirement.

The remainder of the overview should extend the discussion and make clear the intended content and purpose of the process map.

An example of an overview description is given below.

The scope of this exchange requirement is the exchange of information to enable coordination of equipment with other technical design roles, building design and structural design. It includes the need for shape, size and location of components and also for weight to be included.

This exchange requirement allows for the provision of information at various stages during the design process including outline conceptual or sketch design, full conceptual design and coordinated design. The information provided at each stage is essentially the same. However, the level of certainty regarding equipment and components used will increase at each stage allowing greater certainty in space provision.

It is assumed that the information provisions outlined in the exchange requirement er_exchange_HVAC_model (space) have been satisfied. This provides an initial assessment of spaces from the perspective of building design and HVAC design as well as project details.

Information provided through this exchange requirement includes:

- HVAC component type and size
- Shape representation of component type
- Weight of component type
- Location and orientation of occurrences of component type

Example 7: Overview section in exchange requirement

7.3.1.2 Information Requirements

The information requirements provide the breakdown of technical information required by the exchange requirement. It is technical in the sense that it provides the information necessary for technical actions within a project and **not** in the sense that it provides the detailed level of information structure required by a software solution.

The information is provided in a set of information units needed to satisfy the requirement. An information unit typically deals with one type of information or concept of interest such as the overall project, walls, windows etc.

Preconditions

The first information unit described is the set of preconditions for this exchange requirement. A precondition is an action (expressed as an exchange requirement) that must have been completed prior to the execution of the current exchange requirement. That is, exchange requirement 'A' must be completed as a precondition for carrying out exchange requirement 'B'.

Preconditions shall include a description of what should have been done and the name(s) of prior exchange requirements. Note that any exchange requirements that are preconditions of these prior requirements shall be assumed to be preconditions of the current requirement.

An example of preconditions is given below.

Space programming will have been carried out and the area requirements for each space determined. This is so that the total calculated area of the space after configuration can be tested against the space requirement

Prior requirement: er_exchange_space_program

Example 8: Preconditions in exchange requirement

Information Units

Each information unit is then broken down further to provide:

- Name of the information unit
- Description about the information that is exchanged
- Identity of the functional part within which the detailed technical content of this information unit is described
- Attributes or properties that must be exchanged for the provisions of this exchange requirement to be satisfied. Note that only attributes or properties that are mandatory (must be provided) need to be identified within the exchange requirement.
- Any special provisions or rules relating to the attribute or property

An example of a completed information unit is shown below:

Building

Provides relevant information about the building

For technical detail, refer to **fp_model_building**

✓	Placement *This is the placement and orientation of the building relative to the datum point (0, 0, 0) established.*
✓	Building shape *This is strictly optional since it is not a specific requirement for energy analysis but should be available via the same mechanism as the geometry provided for spaces (q.v.). It may however be useful to provide an overall visual context for the space model and is therefore recommended.*
✓	Composition Type *Every building must be defined in terms of its composition type (COMPLEX, ELEMENT, PARTIAL).*

Example 9: Information unit in exchange requirement

7.4 Functional Part

A functional part focuses on the individual actions that are carried out within a business process. An action is concerned with a particular unit of information within an exchange requirement. For instance, to exchange a building model, it is first necessary to model the walls, windows, doors, slab, roof etc. The action of modelling each of these elements is described within a functional part.

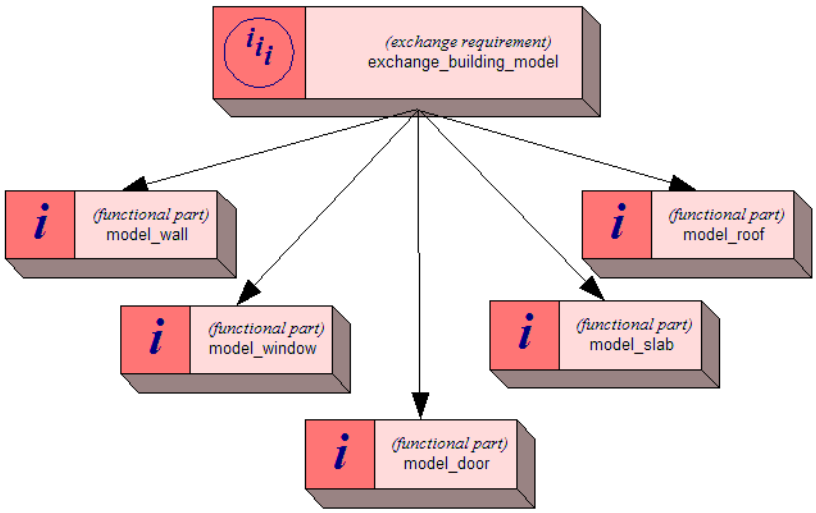


Figure 18: Functional parts in an exchange requirement

Each functional part provides a detailed technical specification of the information that should be exchanged as a result of the action. Since that action may occur within many exchange requirements, it follows that a functional part may also relate to many exchange requirements.

For this reason functional parts are specifically designed to be reusable within many exchange requirements. However, certain functional parts deal with more general ideas and may be expected to participate more frequently. Examples include functional parts dealing with relationships (such as applying a classification to an element) or those dealing with geometric shape representation.

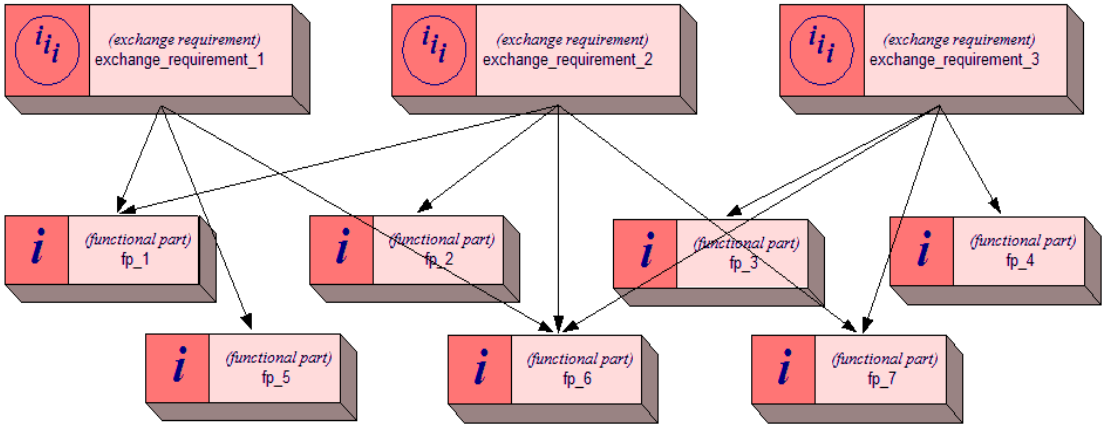


Figure 19: Using functional parts in multiple exchange requirements

This is a very important idea within IDM since it provides the basis for the idea that the technical support of an exchange requirement can be provided by 'shopping' for a basket of functional parts that can then be compiled to provide the exchange requirement model schema.

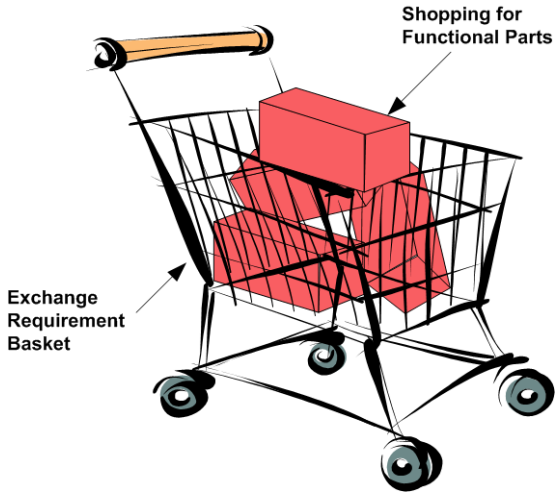


Figure 20: Shopping for functional parts

Functional parts describe an action in close detail. Whereas an exchange requirement describes information in non technical detail, functional parts describe the use of every entity, every attribute, every property set and every property concerned. Because of the detail included, functional parts can also be broken down into other functional parts. That is, a functional part may call on the services of other functional parts in the same way as exchange requirements.

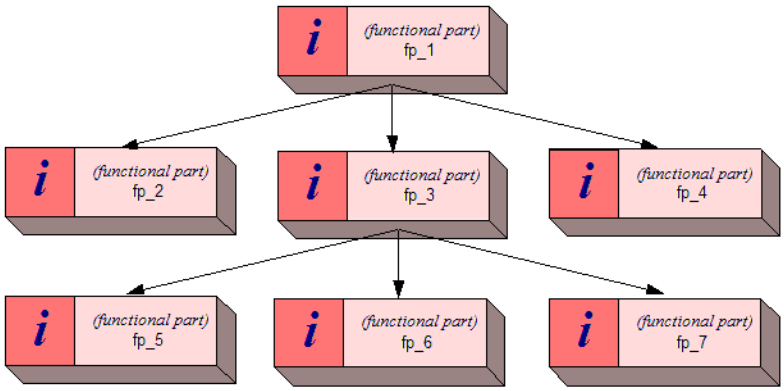


Figure 21: Functional parts using the services of other functional parts

7.4.1 Functional Part Sections

A functional part comprises the following sections:

- header section
- overview section
- results section
- technical section
- lists section
- schema section
- example section

7.4.1.1 Header Section

A header section provides administrative information about the functional part. This includes:

Name	The name or title given to the functional part. This should conform to the IDM naming rules
IFC Release	The identity of the IFC release for which this functional part is valid. Different releases of the IFC model contain revisions to specifications required as a result of usage experience or the inclusion of further capability. Such revisions may affect the content of functional parts.
Identifier	A unique identifier that is assigned to a functional part.
Change Log	Used to mark the creation of the functional part and changes made to it over time. The date of creation/change should be included together with an identifier for the person responsible and a brief description of the changes made.

An example of a completed header section is shown below.

Name	Model System		
Identifier	NBS_FP_0001	**IFC Release**	2x2

Change Log		
2005-07-27	Created	jeffrey.wix@aec3.com
2005-08-20	References to IfcElectricalCircuit added	jeffrey.wix@aec3.com

Example 10: Header section in functional part

7.4.1.2 Overview

The overview states the aims and content of the functional part in non technical text form. The intention is that, whilst a functional part is primarily intended for solution providers, a ‘user’ should still have an awareness of the content since they will be using it in conjunction with an exchange requirement.

The description of an information unit within an exchange requirement should be derived from the overview of the corresponding functional part. Note that this may be either the complete overview or an excerpt.

An example of an overview description from the functional part ' model_system' is given below.

Provides the information concerning systems where a system is a grouping of elements. In the context of a distribution system, all of the elements that are grouped to form the system should be subtypes of IfcDistributionElement.

As well as enabling the description of complete systems, this functional part enables the grouping of elements into subsystems.

For an electrical system, the subtype IfcElectricalCircuit is used. This provides a simple approach to validate that only electrical items are connected to the electrical system.

The information provided about a system includes:

- Specification of a name and description for the system if required
- The elements that are grouped together to form the system

Note that the shape representation of a system is derived from the shape representation of the elements that are grouped together within the system. The system itself has no ‘own’ shape representation.

Further information about the system may be provided through locally defined property sets. There are no predefined property sets for systems.

Example 11: Overview in functional part

It is recommended that the overview section of a functional part includes a graphical description of the concepts that are used in its creation. This should be done using the standard MVD diagramming method.

Since a functional part is bound to an IFC release, the diagram should show concepts that are bound to the same release.

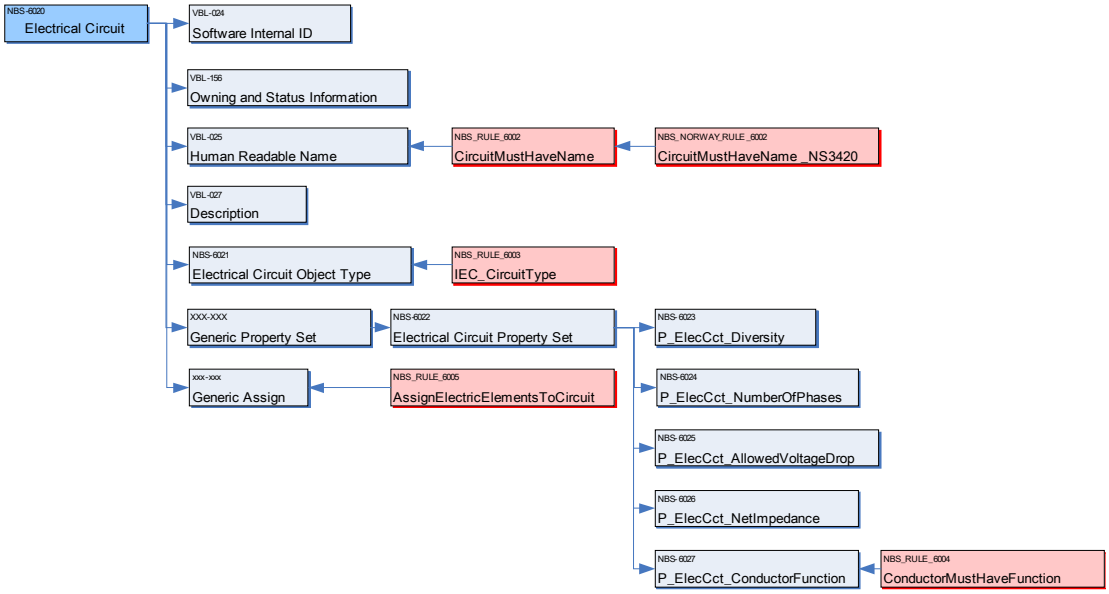


Figure 22: Concept diagram for a functional part

7.4.1.3 Results Section

The results section provides a simple statement of the outcome of a functional part. For instance, the following describes the outcome of a functional part for modelling systems:

Specification of the system and the elements that make up the system.

Example 12: Results in functional part

7.4.1.4 Technical Section

The technical section provides the detailed breakdown of the information provided by the functional part. It describes in detail the IFC entities and property sets required together with the attributes and properties to be used.

The technical section is developed on the basis of a 'flow of events'. That is, it also establishes a reasonable sequence in which the objects/attributes and property sets/properties should be defined. Whilst it is not mandatory for a solution provider to follow this sequence in achieving the functional part, it should provide a good basis from which to work.

Information given within a functional part is presented as a table that includes the following columns:

Description	A detailed description of the information needing to be asserted within the functional part. Each individual data item is described in approximate sequence. An individual data item here is considered to be an object and attribute or a property set and property.
Entity/Pset/ Functional Part	A specification of the actual IFC entity/attribute or property set/property combination that fulfils the information described or a reference to a functional part that provides results into that currently considered. Attributes and properties should also identify the datatype in which they will be expressed. For instance, if the attribute is expressed as a label string, the datatype will be shown as an IfcLabel. The convention defined within the IDM for the expression of object/attribute/datatype and property set/property/datatype is as

	follows: • `Object.Attribute --> Datatype` • `PropertySet.Property --> Datatype` That is: 1. the object name or property set name is given first; 2. a dot (or period) then follows; 3. the attribute name or property name is then given; 4. this is followed by an arrow symbol; 5. the data type is then given. Alternatively, the arrow symbol at step 4 may be replaced by a double colon symbol (::) Note that the definition of a functional part can call on the services of other functional parts for the IFC support required. Note that where an example is provided that describes the use of the particular object/attribute or property set/property, this should be identified. This may also include a pointer to a line in which the description occurs. For example, the reference 'Example1#101' would indicate that further guidance can be found in example 1 at line number 101.
Mandatory/Optional/ Required	An indication of whether the information is mandatory for the functional part (must be provided), optional (may be provided but is not mandatory) or required (optional within the IFC model but considered to be mandatory for this functional part). It should be noted that even though a functional part shows information as optional, a particular usage of the functional part by an exchange requirement might override this and force the information provision to be mandatory. Note that some functional parts may also contain a fourth column that identifies an attribute or property that **MUST NOT** be asserted.

An example of a completed technical section is shown below:

Description	**Entity/Pset/Functional Part**	**MAN**	**REQ**	**OPT**
Specify the system occurrence in which elements will participate *A system opening is directly specified as an occurrence. It does not have a defining type entity.*	IfcSystem OR fcElectricalCircuit	√		
Specify the name of the system. *Although this is an optional attribute within IFC, it must be asserted for system.* *It is strongly recommended that system names are taken from a list of allowable names that have been agreed between organizations that will make use of system information (to ensure that there is common understanding of the system*	IfcSystem.Name --> IfcLabel OR fcElectricalCircuit.Name --> IfcLabel	√		

purpose). The list of allowable names may be taken from a published source such as a classification system or standard reference or it may be agreed in the context of project.				

Example 13: Example of technical description in functional part

7.4.1.5 Lists Section

The list section of the functional part provides an annotated list of the various IFC components used. These include:

Entities	The subjects (or classes of item) of interest within the current functional part
Datatypes (defined, enumeration and select)	Named types of data that may be used within the functional part such as labels, text descriptions, identifiers OR; Enumerated ranges of possible values from which a selection should be made OR; Alternative selections of route through the schema
Function	Extended rules that may be processed to validate data (such as determining that a particular date is within the legal range of possible values for that month and year)
Property sets	Those property sets that are relevant to the current functional part
Functional parts	Other functional parts whose services are used within the current functional part
Propositions	Additional rules and functions that are applied within the context of this functional part in addition to those that form part of the complete IFC schema. Propositions relating to particular items are shown in the technical section; this section provides a reference list.

7.4.1.6 Schema Section

The schema section provides the formal description of the IFC capabilities used. This section is specifically for solution providers to guide their implementation development. It is not necessary for users to read this section.

There are various presentation forms that may be used to describe a schema. The principal forms used by buildingSMART® for the presentation of IFC are EXPRESS-G and EXPRESS. Alternative forms include ifcXML.

EXPRESS-G Graphical Form

EXPRESS-G is a graphical notation that is used in the development of IFC. It has the advantage of being relatively easy to understand and use. The formal definition of EXPRESS-G is contained within ISO 10303 Part 11.

An EXPRESS-G presentation shows the entities, attributes and data types within a schema. A 'guide to EXPRESS-G' is provided as part of the buildingSMART® document set.

Each functional part schema within the IDM is intended to be a complete schema in its own right. It is a subset of the complete IFC schema and is effectively a 'cut out' version of those parts of the IFC schema that are of current interest. Entities and attributes are defined in exactly the same way as in the full IFC schema. The benefit to a solution provider is that instead of having to identify relevant IFC capabilities from the complete schema with its hundreds of entities, the capabilities needed for the required function and how the function should be achieved are explicitly identified within the functional part.

A functional part may also use other functional parts. The usage of called functional parts in IDM differs slightly from existing capabilities provided within the EXPRESS-G notation. For this reason, IDM uses a custom addition to the EXPRESS-G notation to define functional parts used and to ensure that they appear in the more complete EXPRESS schema (see Appendix 'IDM EXPRESS Creation Guide').

Property sets used by a functional part do not appear within the EXPRESS-G presentation of a functional part. This is because they are user defined extensions. Property sets are identified in the list section described above.

EXPRESS Language Form

EXPRESS is the data definition language in which the primary form of the IFC schema is developed. It is a powerful language in which to define data structures and data validation rules. The formal definition of EXPRESS is contained within ISO 10303 Part 11. A 'guide to EXPRESS' is provided as part of the buildingSMART® document set.

Each functional part schema a schema is presented in the EXPRESS language. This is a true subset of the complete IFC schema and contains not only the entities and functional parts that are visible in the EXPRESS-G form of the schema but also validation (WHERE) rules, derivations of data values, unique attributes on entities and functions for determining and validating data values.

Functional parts schemas may contain rules that are additional to the full IFC schema. Such rules may be used to provide additional control on the data content of a functional part and respond to the propositions identified within the list section (see above)

The EXPRESS language form of the functional part schema is the normative form.

ifcXML Form

ifcXML is the version of the XML syntax that can be automatically generated from the IFC EXPRESS schema using the XML Language binding defined within ISO 10303 Part 28.

Functional part schemas are defined in ifcXML.

The ifcXML language form of the functional part schema is an informative form.

Other Forms

As appropriate, other forms of schema representation may be added to functional part specifications.

7.4.1.7 Example Section

The examples section includes particular examples that show how the functional part might be used. It is useful to provide more detailed guidance to implementers and may be used by them for preliminary testing to ensure that they are returning the correct results from their solutions.

Examples may include:

- a description of the example scenario
- samples of IFC files according to the schema of the functional part to show how the results of using the functional part might appear in practice
- sample instance diagrams providing an easier visual reference

Example 1: Definition of a single geometric representation context, being the basic model context. It only contains the absolute mandatory information.
Note: It is an IFC2x2 example which still has the WorldCoordinateSystem given, even if it defaults to Location=[0.,0.,0.] an z-axis=[0.,0.,10.] and x-axis =[1.,0.,0.].

```
/* Definition of the project */
#17=IFCPROJECT('02b_zn_n5BUhvEFQj1tiGU',#16,'DefaultProject','Automatically generated
project',$,$,$,(#11),#6);
```

```
/* Definition of the geometric representation context */
#11=IFCGEOMETRICREPRESENTATIONCONTEXT('TestGeometricContext','Model',3,$,#10,$);
#10=IFCAXIS2PLACEMENT3D(#9,$,$);
#9=IFCCARTESIANPOINT((0.,0.,0.));
```

Example 14: Example within a functional part

Figure 23: Example instance diagram

7.5 Exchange Requirement Model

An exchange requirement model is the technical solution of an exchange requirement. It provides a complete schema that can be supported by a software application for the exchange of information for a particular purpose, at a particular point in time on a project and at a particular location. That is, it satisfies all the conditions for supporting a project workflow according to the rules and methods of working defined for a region, country or framework agreement.

An exchange requirement model is a specific technical solution for an exchange requirement. However, the exchange requirement model is dependent of the release version of the information model from which it is derived. Therefore, an exchange requirement may have several exchange requirement models as technical solutions, each technical solution supporting a particular release of the information model.

7.5.1 Compiling Exchange Requirement Models

For exchange requirements (and therefore exchange requirement models) and functional parts, IDM provides a recursive structure. This means that an exchange requirement can include other exchange requirement (other than itself) and a functional part can include other functional parts (other than itself).

The principal relationship however is that an exchange requirement is compiled from the functional parts that have been declared to support the exchange requirement. Practically, this means that the schema (or data structure) of an exchange requirement model is developed by adding together the schemas of the set of included functional parts.

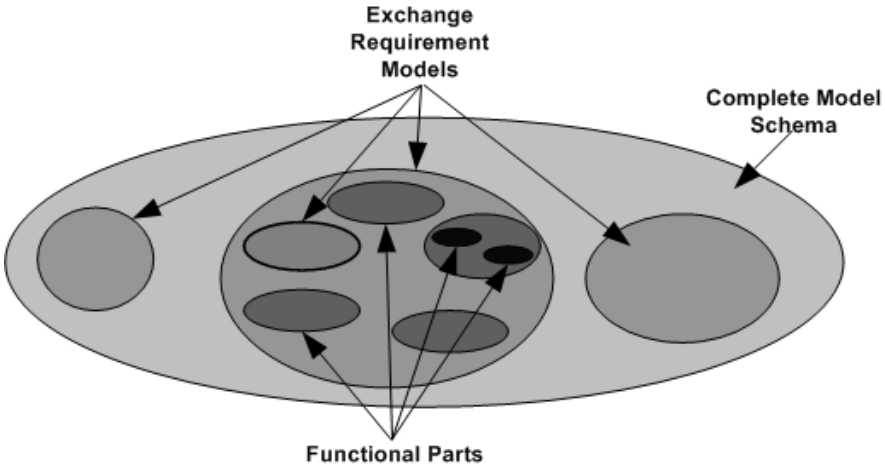


Figure 24: Hierarchy of components in IDM

Including Functional Parts

The principal unit of the IDM in which a schema is expressed is a functional part. This defines technical content.

Each functional part has a fully developed schema. However, the schema of a functional part can call upon, or include, the schema of other functional parts.

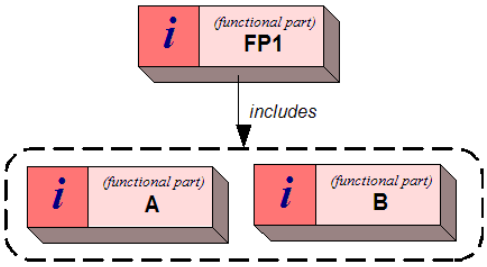


Figure 25: Functional part includes other functional parts

Consider the functional part FP1 which contains the entities A and B. The schema for this functional part could be described as being the integration (sum) of the contained entities:

$FP1 = \sum i$ (where i stands for any entity)

Which expands to:

$FP1 = A + B$

If we now consider another functional part FP2 which is contained within the functional part FP1. FP2 contains the entities C and D. Therefore:

$FP1 = A + B + FP2$ and $FP2 = C + D$

Thus:

$FP1 = A + B + C + D$

The nature of compiling a functional part schema by addition however means that there may be overlap in the entities used in each schema. That is, simply adding the entities into the total schema could create a situation where the same entity is included more than once. This is not allowed. Each entity may only appear once in the total schema. Therefore, the process of adding schemas together

also has to resolve the issue of overlapping entities such that each required entity appears only once. That is:

```
if the schema FP1 contains the set of entities {S, X}
and FP2 = SET {U, V, W, Z}
and FP3 = SET {T, U, V, X, Y}
then FP4 = SET {S, T, U, V, W, X, Y, Z}
```

Schematically, this is shown in the illustration below.

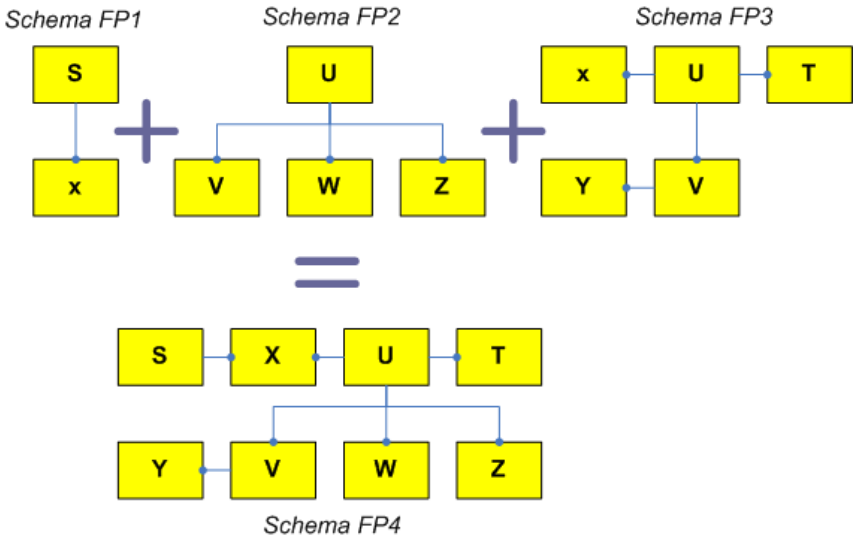


Figure 26: Resolving entities added from schemas

Since an exchange requirement model is made up of functional parts, it follows that the schema for the exchange requirement model is determined by adding together the schemas of the contained functional parts in the same manner as above. That is:

```
ERM1 = FP1 + FP2
```

Since FP1 = A + B + C and FP2 = D + E + F then it follows that:

```
ERM1 = A + B + C + D + E + F
```

This is illustrated below.

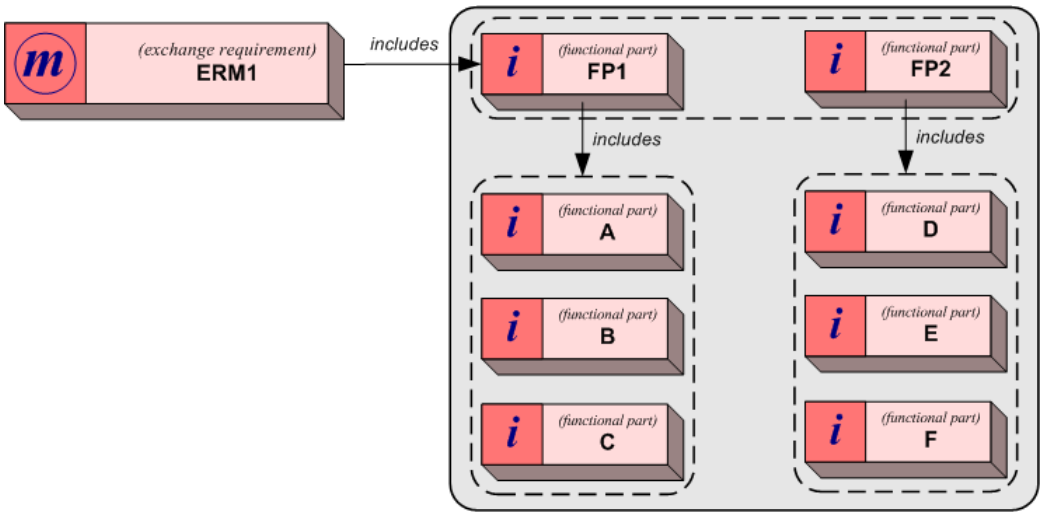


Figure 27: Decomposition of functional parts

Including Exchange Requirement Models

In the same way as an exchange requirement model includes functional parts, it can also include other exchange requirement models. That is, information already defined for a prior exchange requirement model may be used within a current exchange requirement model. This facility can be used effectively to reduce the effort needed in describing the information requirements. That is:

```
ERM2 = ERM1 + FP1 + FP2
```

In fact, using an exchange requirement model in this way is no different to the addition of functional parts above. This is due to the fact that the exchange requirement model ultimately reduces to the functional parts from which it is built. Therefore, a reference to an exchange requirement model included is only a reference to the set of functional parts from which it is compiled.

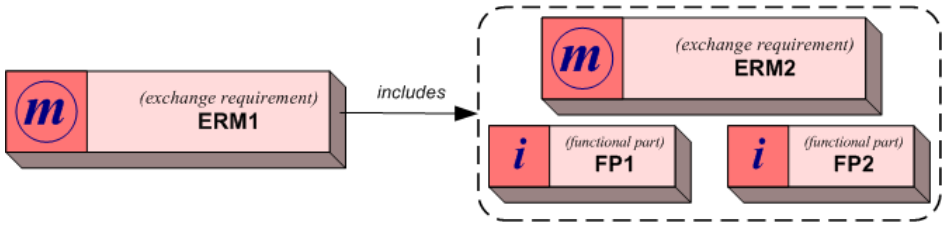


Figure 28: Including exchange requirement models and functional parts

7.6 Business Rules

In IDM, a technical expression of an exchange requirement can be derived by compiling the constituent functional parts. This defines a coherent schema that can be used as a specification for information exchange within the scope of the exchange requirement.

However, IDM also seeks to provide a finer degree of control over information exchanged such that the schema can be applied in national, local or even project contexts. It does this through the provision of business rules that can act on the content of an exchange requirement model.

The primary intention of a set of business rules is to enable an exchange requirement model to be modified and tailored to a specific business need. For instance, a generic exchange requirement model may be developed for a particular purpose (e.g. cost modelling at the detailed design stage of a project). Business rules can configure this requirement so that it is specific to a place without having to

actually change the schema underlying it. For instance, two different sets of business rules might be applied to a single exchange requirement model to e.g.:

- cost modelling at the detailed design stage of a project in UK
- cost modelling at the detailed design stage of a project in Norway

7.6.1 Business Rules Sections

Business rules sets may be either generic (i.e. generally applicable in an international context) or local (i.e. extends a generic business specification so that it is applicable in a local context). It is also possible to extend local business rules with sub local rule extensions.

A business rules set (generic or local) comprises two sections:

- header section
- rules table

7.6.1.1 Header Section

A header section provides administrative information about the business rules. This includes:

Name	A name given to the set of business rules. This should be used to identify purpose of the rule set and the place at which it is applicable
Extends	Name of the higher level business rule set that is extended. Note that if the location given for this business rules set is 'Generic', this is a top level set and this item should be left blank.
Applicable to ERM	The exchange requirement model to which the business rules should be applied.
Location	The place at which the business rule set is applicable. For a top level business rule set that is generally applicable, the value 'Generic' should be used.
Identifier	A unique identifier that is assigned to a business rule set.
IFC Release	The IFC release to which the business rule set applies. This is derived from the exchange requirement model to which the business rules apply.
Change Log	Used to mark the creation of the business rule set and changes made to it over time. The date of creation/change should be included together with an identifier for the person responsible and a brief description of the changes made.

An example of a completed header section is shown below.

Name:	**br_exchange_electrical_system**	***Extends:***	
Applicable to ERM:	**erm_exchange_electrical_system**	***Location:***	**Generic**
Identifier	NBS-BR6001	**IFC Release**	IFC 2x3g

Change Log		
2007-10-21	Created	jdw@aec3.com

Example 15: Header section in business rules

7.6.1.2 Rules Table

The rules table provides specific detail about the business rules that are to be applied.

Each rule is given an identifier. This should reflect the enterprise, activity, organization or individual responsible for its development together with the location (if appropriate) at which it is applicable.

Each rule is also given a name that is expressive of its purpose.

The proposition of each rule should be expressed textually. Propositions should contain informative notes and guidance for the solution provider where necessary. This can include guidance on how a user might expect to apply the data or any agreements made between solution providers about how the data item should be interpreted.

Rule id	Name	Proposition	Allowed Value(s)
NBS-RULE-6001	AggregateElectricElementsToDistributionPoint	Only occurrences of particular types of electrical distribution elements may be aggregated within a distribution point.	IfcFlowController --> IfcElectricTimeControlType IfcFlowController --> IfcSwitchingDeviceType IfcFlowController --> IfcProtectiveDeviceType IfcFlowTerminal --> IfcOutletType
NBS_RULE_6002	CircuitMustHaveName	An electrical circuit must have a name. At the generic level, any name assertion will satisfy this rule	EXISTS IfcElectricCircuit.Name = TRUE

Example 16: Rules section in business rules

7.6.1.3 Business Rules in MVD Diagrams

Whilst business rules operate at the level of the exchange requirement model, in practice they act on concepts. This fact can be used to enable business rules to be graphically established in the concept/functional part diagrams. This is shown in figure 29 below.

The rule identifier and rule name are shown in the rule box. The rule box itself has a different color to that used for generic or bound concepts. The connection arrow is drawn from the rule to the concept to identify that the rule acts on the concept and not vice versa.

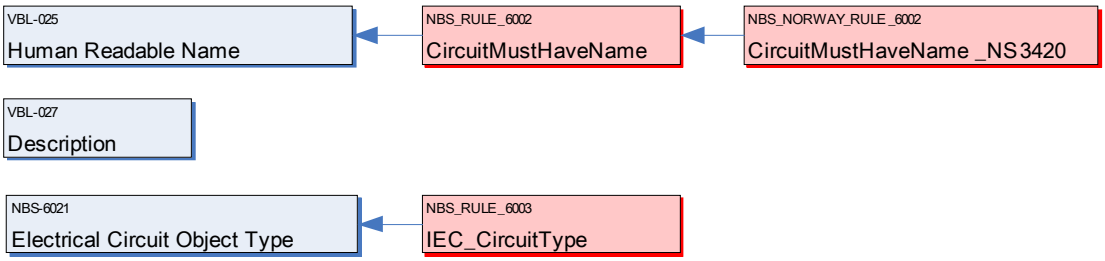


Figure 29: Specifying rules in an MVD diagram

8. IDM Development Process

The IDM development process has the objective of understanding business requirements for information exchange and/or sharing and developing an object model and software implementations that can be used by practitioners within the industry to satisfy those requirements. It is a process that results in a number of components whose interaction is examined in the IDM Technical Architecture section (see above)

A basic picture of the relationship of the various components described by the technical architecture and the development flow can be described by Figure 30 below. This illustration is termed the 'Exchange Requirements Factory' since it places the exchange requirements at the heart of the development process and considers all other components as supporting this objective.

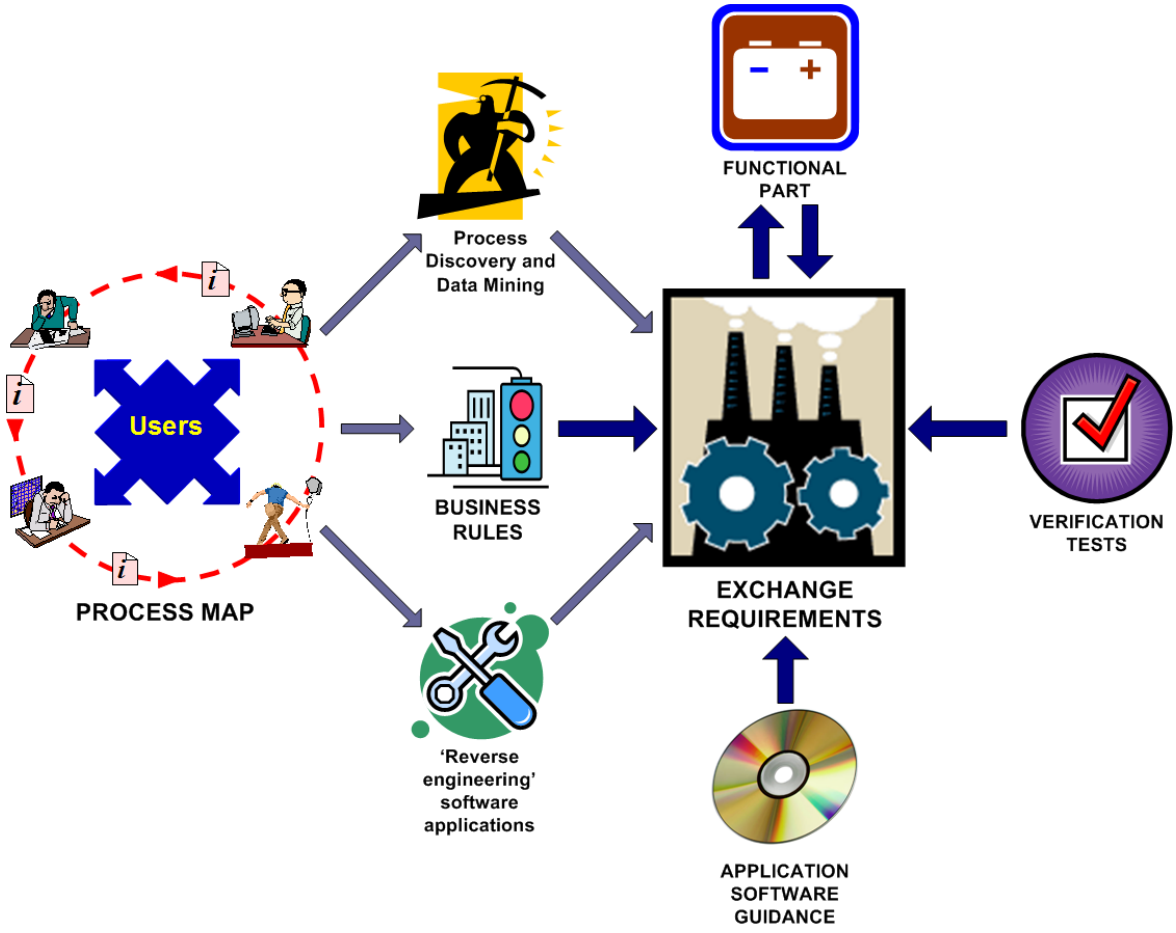


Figure 30: Exchange Requirement Factory

8.1 Propose IDM

A proposal to undertake an IDM development is a preliminary stage that sets the scene for work to be done. It is concerned with:

- defining the scope
- identifying and allocating resources
- fixing the development route
- establishing a project development plan

There is no formal organizational process for approving or authorizing an IDM development. The whole concept of IDM is that it is completely free and open to groups who determine that they have a need for an IDM development to simply go ahead and do it.

However, it is a good idea to 'register' the fact that IDM development is progressing with the IDM technical team, the IDM web site and others so that:

- skills and guidance from the IDM technical team and other IDM development projects can be accessed
- awareness of other IDM development projects local to place can be obtained
- awareness of other IDM development projects in a similar technical domain but within other places can be obtained

8.1.1 Scope

A proposal should identify the scope of the IDM development. The scope should set the boundaries for the work that is to be done and provide a continuing reference to ensure that the work boundaries do not grow beyond a point at which the planned or available resource ceases to be sufficient.

8.1.2 Identify Resources

Resources are the people and components that will bee used in the IDM development. Resources need to be properly balanced between project management, technical execution, industry knowledge and solution provision. The required balance of resources will be affected by the development route that is selected.

Resource identification should also take account of any hard costs that need to be consumed within the development and ensure that the availability of finance for such costs is sufficient to ensure that the development scope can be met.

8.1.3 Fixing the Development Route

The development route will be determined by the development approach selected and this, in turn, will impact on the required resources. Development routes are described below.

8.1.4 Project Plan

The project plan is simply the management aid that should be used to assess the period over which the development is to occur, determine the tasks, assign the available resources and set the deliverables required. Managing an IDM development is not expected to be significantly different to managing other types of projects. Some of the technical execution skills will be different however.

8.2 Develop IDM

IDM development comprises creating the various components that collectively make up an 'IDM'[3]. This includes the process map, exchange requirements, functional parts, business rules and concepts.

There are three commonly used routes to IDM development currently identified (others may also exist). These are shown in the 'Exchange Requirements Factory' diagram and can be termed:

Process discovery and data mining

Business rule localization

Reverse engineering

8.2.1 Process Discovery and Data Mining

Process discovery and data mining is the conventional process used in IDM development and also in IFC development and extension. It assumes that there is no initial presence of software or other exchange requirements.

The development approach is described below as a linear sequence. In practice, feedback between development stages and cyclic developments can be expected.

3 Strictly speaking, an IDM development is about the development of a set of components as described above. However, the complete package of components is sometimes referred to as an IDM

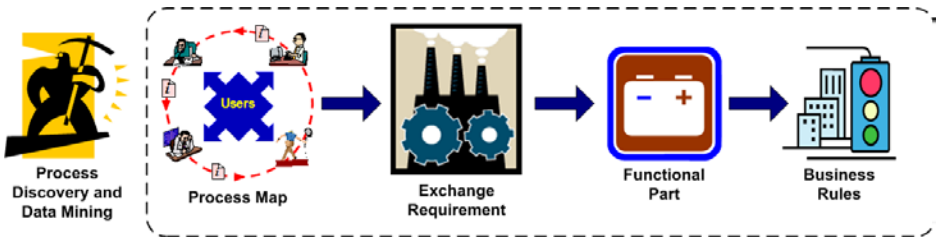


Figure 31: Process discovery and data mining development sequence

- **Discover Process**
  This involves working primarily with industry specialists (which can also include specialist solution providers) to determine the business processes within the scope that need to be satisfied.

  Process maps can be created either directly by the industry specialists or, more probably, a modelling specialist eliciting the relevant processes from specialists.

  Process map discovery usually requires several cycles of development and review to achieve a satisfactory conclusion. On completion, the process map may represent the business process as currently practiced or it may represent a proposal for an improved business process. Whether to create an 'as-is' process or a 'to-be' process is a conscious decision that needs to be taken by the development team.

  Within the process maps that are developed, the exchange requirements that are necessary will be identified.

- **Mine Data**
  When locating the exchange requirements within the process map, the data requirements should also be specified using the industry specialist knowledge. Initially, this may be just a list that accompanies the process map documentation but it provides the keys to exchange requirement and functional part development.

- **Create Exchange Requirement**
  Exchange requirements should then be created according to the format described in section 7.3.

  Step 1 is to identify the exchange requirement

  Step 2 is to create the overview

  Step 3 is to determine if functional parts that satisfy the needs of the exchange requirement have already been developed. If so, their overviews (or overview excerpts) should be imported to the exchange requirement and the mandatory data provision explained.

  Step 4 occurs if functional parts do not exist. In this case, they need to be created so that the provisions of step 3 can then be achieved.

- **Create Functional Parts**
  Functional parts identified as needed should then be created according to the format described in section 7.4. This is a technical task that should be carried out by a solution provider or modelling specialist.

- **Define Business Rules**
  The set of business rules that may need to be applied over and above the exchange requirements/functional parts should be defined. These may be used to determine attributes/properties to be asserted or to control values that may be given to attributes/properties.

  Sets of business rules should be named and created according to the format described in section 7.6.

8.2.2 Business Rule Localization

Business rule localization assumes that an exchange requirement exists for the purpose required but that it does not mean the specific needs for use within a particular location. Location may be a place (country, region etc.), a project or a framework of working agreed between organizations.

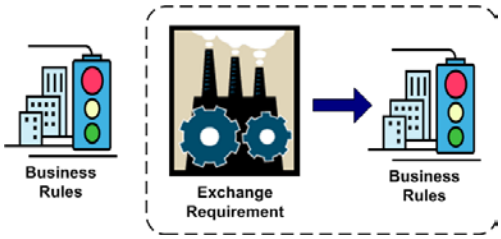


Figure 32: Business rules development sequence

Business rule localization also assumes that the process map within which the exchange requirement is defined exists and that all functional parts that support the exchange requirement are defined.

All that is required in this case is that the business rules that may be applied to make the exchange requirement specific to a location (where location is as described in 'define business rules' for process discovery and data mining above).

Business rule definition needs to be collaborative between industry specialists who know the data requirements in that location and technical specialists who can express the business rules in a form that makes them applicable in conjunction with the information exchange schema underlying an exchange requirement.

8.2.3 Reverse Engineering

Reverse engineering assumes that software already exists that is capable of dealing with the information exchange(s) required but that there is a need to specifically capture from that software the exchange requirements that it can support. This also makes a prior supposition that a 'model view definition' exists and that the software concerned has been certified against it or that there is an intention to create a model view definition by integrating the content of the exchange requirements supported and then certifying the software.

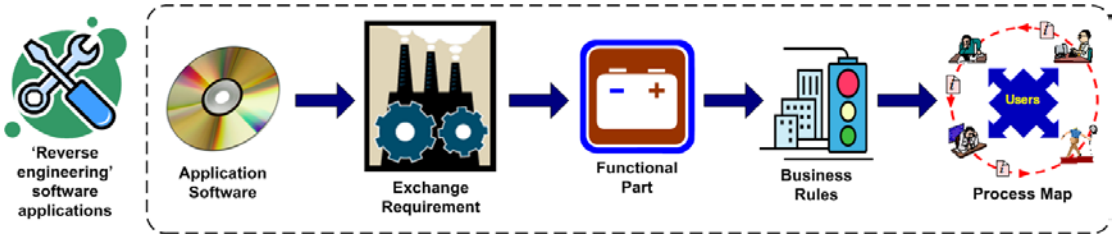


Figure 33: Reverse engineering development sequence

In reverse engineering, the starting point has to be the software application(s) concerned. This may be one particular software application (in which case the assumption is that the exchange requirement will probably be localized) or several similar software applications (in which case a more general exchange requirement case may be concerned).

The most appropriate way of doing reverse engineering is to define the scenario that the exchange requirement is to support and then to work through the software application to provide the data. The dialogue boxes that describe the data provide the clues.

Reverse engineering is typically about providing data to the application that you are working with. That is, you are trying to acquire input data in the most economic manner.

Thus, for each data item that you have to complete in the dialog box, the question should be asked 'could this be easily acquired by data exchange from an upstream application'?

An example of this might be in the specification of length, width and height of a space. These should all be available through the definition of a bounding box within a geometric BIM application. Therefore, these data items belong in the exchange requirement.

- **Define scenario**
  Define the scenario that the exchange requirement is to support. This should provide a detailed textual description that can be used as the overview description for the exchange requirement.

- **Recover data**
  Working through the defined scenario within the software application, recover all of the data that needs to be specified to achieve a result within the application.

  For each data item recovered, determine if it could be acquired from an upstream software application. If so, it should form part of the exchange requirement.

- **Create Exchange Requirement**
  The exchange requirement should now be created using the defined scenario as the overview and the identified data within the technical sections.

  Data items should be checked to determine if functional parts exist that satisfy the specific needs of the exchange requirement. If so, their overviews (or overview excerpts) should be imported to the exchange requirement and the mandatory data provision explained.

  If functional parts do not exist, they need to be created so that the data provisions of the scenario can be achieved.

- **Create Functional Parts**
  Functional parts identified as needed should then be created according to the format described in section 7.4. This is a technical task that should be carried out by a solution provider or modelling specialist.

- **Define Business Rules**
  The set of business rules that may need to be applied over and above the exchange requirements/functional parts should be defined. These may be used to determine attributes/properties to be asserted or to control values that may be given to attributes/properties.

  Sets of business rules should be named and created according to the format described in section 7.6.

- **Capture Process**
  As one or more exchange requirements are reverse engineered from software applications, they can be captured in a process map. The process map may be generated from the capabilities of the software application or an existing process map may be modified.

8.3 Define Terminology

As IDM development proceeds, the terminology that is applied within the business process should be captured. This can then be provided with the IDM components (exchange requirements or process map) or, alternatively, entered into a standard dictionary such as an IFD based dictionary[4].

8.4 Publish IDM

As IDM components are completed, they should be published at the international IDM web site in the Confluence Wiki format as outlined in this methodology.

8.5 Generate and Manage Issues

IDM developments should be reviewed by industry experts, solution providers and information modellers other than those who initially created them for the following reasons:

- ensures that components are validated by the agreement of a number of industry experts;
- identifies aspects of the components that have not been fully included;
- ensures that the components are applicable at the appropriate level (international, national, project etc.);
- allows for the correction of inaccuracies.

4 This is the preferred solution. Capture and entry to the dictionary can be achieved using the IFD web services API. Refer to the IFD library guidance at www.ifd-library.org for further information/

The following table identifies typical reviewers, what they should review and their purpose in doing so.

You are …	*You should review …*	*To determine …*
A manager (either within AEC/FM or IT applications)	The process map, and the overview statement of the exchange requirement	Whether the IDM development meets your business needs
An industry expert, or a modeller responsible for developing specifications and systems that support business processes	The process map, exchange requirement (complete) and business rules statements	Whether all the data that you use for the business process is completely and correctly identified
An applications software developer, or a modeller involved in designing and developing systems	The process map, exchange requirements (complete), functional parts, busoiness rules and concepts	How the IDM development can be mapped to your application's data, or to the data held in your database(s)

Reviews may generate issues about the IDM development that should be recorded within the IDM issues resolution database. This may be accessed through https://www.hawkis.com/jira.

To generate issues on the database, appropriate user rights should be requested from the IDM Technical Team. Information on applying for this is given at the IDM web site (http://idm.buildingsmart.com)

9. Using IDM for application development

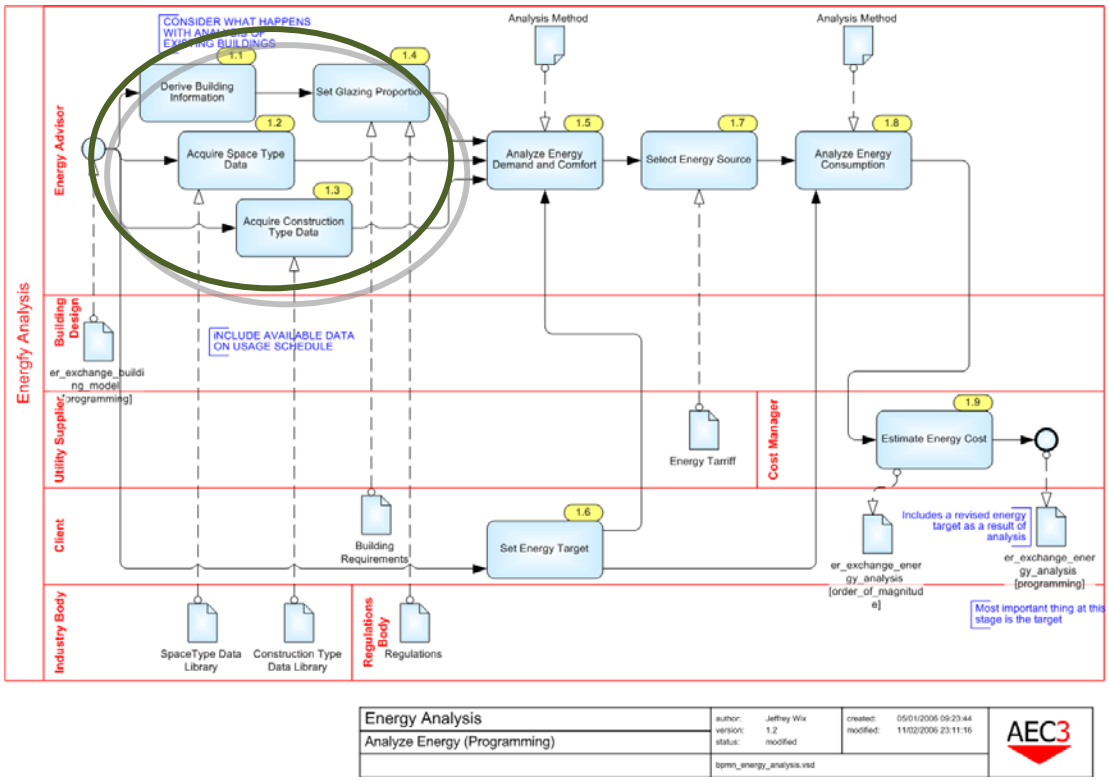
The end result of IDM development is a software application (or applications) that can exchange information according to the protocols established by the exchange requirements and at the stages in the process that have been identified.

This section offers guidance on IDM implementation. It is aimed towards the application developer, but should also be of interest to practitioners looking for improvements in the software tools available to them.

9.1 Defining the scope

Typically, an application can address several discrete steps in the process diagram and may be able to address the activities of more than one role. In developing an application to support IDM, the first step is to identify the scope of the intended implementation. It may address a large part of the process as shown on the process map or it may address several distinct process areas. Even if an application addresses a large area, it may be beneficial to break up the functionality, recognizing that some areas may in practice be addressed by other actors or other applications. Each 'bubble' drawn on the process defines a specific scope.

- Can the application support all the activities within the scope?
- Which of the (swimlane) actors will the application serve?
- What applications may be serving the remainder?



The application must provide interoperability across the boundaries of the scope, where the arrows cross. This may involve reading and writing external resources, as well as interacting with the user to obtain and publish extra information.

If an implementer is involved with a specific project, there may be a specific 'project agreement' about how to deal with information and develop applications that support it. Practically, this will often be developed instead of a process map; however, it is strongly recommended that a process map should be developed so that all information exchange activities scan be seen in context. A project agreement lists the detailed exchanges that have been agreed for a project. Without a process map however, it may be difficult to identify which of the detailed exchanges can be implemented.

9.2 Extracting the relevant IDM components

At each point where the scope boundary is crossed by a 'message' flow, an 'exchange requirement' may be derived. A schema, or information specification, can be derived for the exchange requirement by compiling the schemas of the contained exchange requirements and functional parts (see section 7.5).

This exchange requirement schema can be used as the specification for a file based information exchange between two separate software applications or it can be used for partial model exchange using a model server.

Practically, for software implementation, it is expected that all the exchange requirements that an application is intended to support will be used together to generate the 'schema' (or software specification) that will guide the software implementation. This is substantially the same as a 'model view' (see section 6)

For IDM implementation, the IDM schema derived will need to be compared to and aligned with the internal schema of the application. This is done by mapping the IDM requirements to the software application capabilities.

9.3 Mapping entities

The IDM schema identifies the entities involved in the exchange. These are mapped to/from the representation of the entities that is used by the application. In many cases the correspondence is obvious. However, there will also be cases where the correspondence may be complex. The means of representing and transforming needs to be determined. Some concepts, e.g. 'envelope', may not have a direct representation. In this case algorithms may be needed to define the mapping.

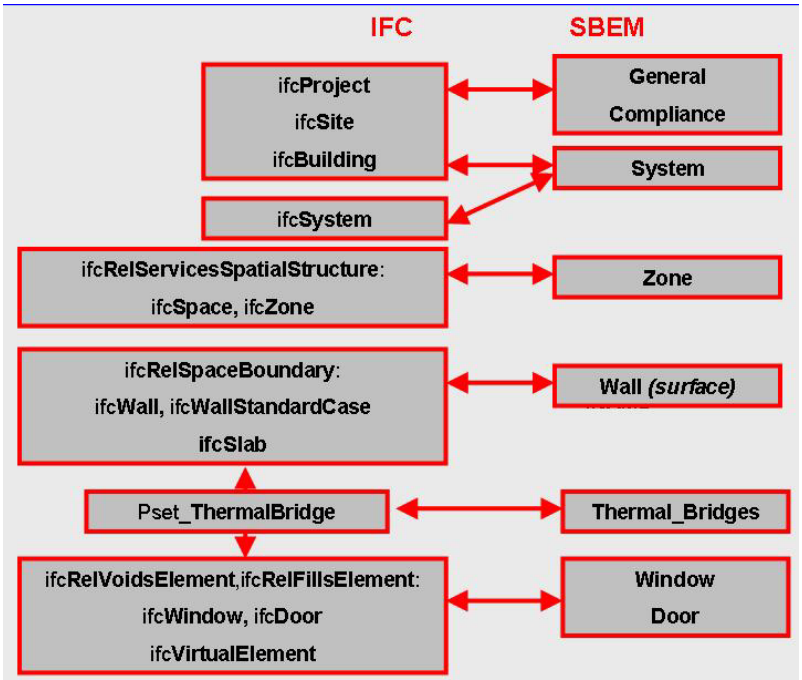


Figure 34: Mapping IFC entities and property sets to the SBEM software

In the example shown in Figure 34, a thermal analysis application (the SBEM application for UK building regulations part L2 compliance checking) reads an IFC dataset. Some entities are easily aligned. However the SBEM 'Wall' concept is not obvious since it is mapped from ifcSpaceBoundary.

9.4 Mapping attributes

The IDM schema identifies the attributes required in the exchange. These are mapped to/from the representation of the attributes that is used by the application. In many cases the correspondence is obvious. However, there will also be cases where the correspondence may be complex. The means of representing and transforming needs to be determined. Attributes that are required by a software application but that have not been considered within the overall IFC schema may need to be stored into, or read from, extension 'property sets' created for the purpose..

In the example shown in Figure 35, the SBEM application reads an IFC dataset. Some attribute values are derived from the dataset and some have to be supplied.

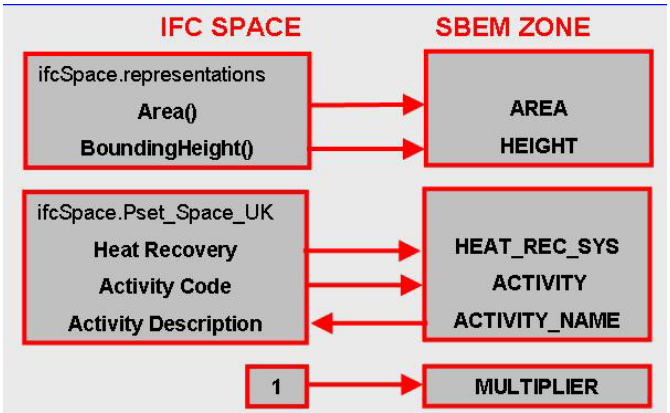


Figure 35: Mapping IFC attributes to the SBEM software

9.5 Mapping constraints

Constraints can be used to validate the exchange. Most constraints demand that an attribute that is optional in the full IFC schema, is either mandatory, or has a limited set of allowed values in the context of the particular exchange requirement.

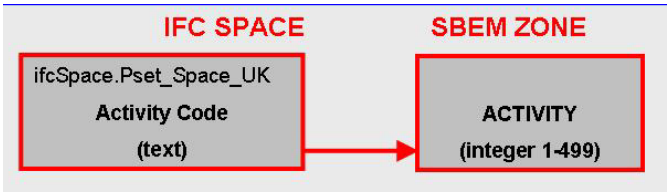


Figure 36: Mapping constraints to the SBEM software

9.6 Validation

Exchange requirements can be represented within a model repository, along with functional parts and business rules. If the business rules have also been represented in an interpretable form, then any dataset can be validated against the requirements.

9.7 Compliance

Any application can be checked for compliance with the nominated views (sub-schemas) of the IFC model. These may be published informally as test-criteria or increasingly as 'Model View Definitions' (MVDs) in the format approved by the buildingSMART® . To achieve certification, an application must demonstrate a high degree of pass-through if the application is file based, and a high standard of data validation if it is server based.

10. References

Business Process Modeling Notation Specification, OMG Final Adopted Specification, OMG, 2006, available at: http://www.bpmn.org/

Extensible Markup Language (XML) 1.0 (Fourth Edition), W3C Recommendation 16 August 2006, edited in place 29 September 2006, W3C, 2006, available at: http://www.w3.org/TR/xml/

Function Definition Modeling Using IDEF0, Federal Information Processing Standard 183, National Institute for Standards and Technology, 1993

Gehre A., Katranuschkov P., Wix J., Beetz J., *InteliGrid Deliverable D31: Ontology Specification,* The InteliGrid Consortium, c/o University of Ljubljana, 2006, available at: http://www.InteliGrid.com

Hietanen J., *IFC Model View Definition Format,* International Alliance for Interoperability, April 2006, available at http://www.iai-international.org/software/MVD_060424/IAI_IFCModelViewDefinitionFormat.pdf

IFC 2x Edition 3 Model (IFC 2x3) Release, International Alliance for Interoperability, available at http://www.iai-international.org

ISO 10303-11:2004, Industrial Automation Systems - Exchange of Product Model Data - Part 11: Implementation Methods; EXPRESS Language Reference Manual, ISO, 2004

ISO 10303-21:2002, Industrial Automation Systems - Exchange of Product Model Data - Part 21: Implementation Methods; Clear Text Encoding of the Data Structure, ISO, 2002

ISO 12006-2:2001, Building Construction -- Organization of Information about Construction Works -- Part 2: Framework for classification of information, ISO, 2001

ISO/FDIS 12006-3:2006, Building Construction -- Organization of Information about Construction Works -- Part 3: Framework for object oriented information, ISO, 2006

ISO/IEC 19501:2005(E), Information Technology - Open Distributed Processing - Unified Modeling Language (UML) Version 1.4.2, ISO, 2005, available at: http://www.omg.org/docs/formal/05-04-01.pdf

ISO/DIS 22263, Building Construction -- Organization of information about construction works --Framework for project management, ISO, 2006

ISO 29481-1:2010, Building information modelling -- Information delivery manual -- Part 1: Methodology and format

Kagioglou M., Cooper, R., Aouad, G., Hinks, J., Sexton, M., Sheath, D. M., *A Generic Guide to the Design and Construction Process Protocol*, University of Salford, UK, 1998, available at: http://www.salford.ac.uk/gdcpp/

Katranuschkov P., Wix, J., Liebich, T., Gehre, A .; *Collected End User Scenarios*; Deliverable D11, EU Project IST-2001-33022 (ICCI); Technical University of Dresden, 2002

Katranuschkov P., Wix, J., Liebich, T., Gehre, A.; *Collected End User Requirements and Common Structure Part I: Description of Methodology and Requirements Synthesis* ; Deliverable D12-1, EU Project IST-2001-33022 (ICCI); Technical University of Dresden, 2002

Katranuschkov P., Wix, J., Liebich, T., Gehre, A.; *Collected End User Requirements and Common Structure Part 2: Gap Analysis* ; Deliverable D12-2, EU Project IST-2001-33022 (ICCI); Technical University of Dresden, 2002

OmniClass Construction Classification System, Version 1.0, CSI/CSC, 2006, available at: http://www.omniclass.ca/

Sparx Systems UML Tutorials, *The Business Process Model*, Sparx Systems, available at: http://www.sparxsystems.com.au/downloads/whitepapers/The_Business_Process_Model.pdf

Unified Classification for the Construction Industry, RIBA Publications, 1997

White S.A., *Introduction to BPMN*, IBM, 2005, available at: http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf

Wix J. ed, *Information Delivery Manual: Using IFC to Build Smart,* 2005, available at: http://www.iai.no/idm/learningpackage/idm_home.htm

Wix J. ed, *IFC Specification Development Guide Edition 2, 1998*, available at: http://www.iai.no/documents/IFC_R2_SpecificationsDevelopmentGuide_A4.pdf

Wix J., Karstila K., buildingSMART Glossary of Terms, 2007, available at http://www.iai-tech.org

Wix, J., Liebich T. , *Information Flow Scenario*, Deliverable D4, EU Project IST-1999-11508 (ISTforCE), Technical University of Dresden, 2000

XML Schema 1.1 Part 1: Structures, W3C Working Draft 31 August 2006, W3C, 2006, available at: http://www.w3.org/TR/xmlschema11-1/

XML Schema 1.1 Part 2: Datatypes, W3C Working Draft 17 February 2006, W3c, 2006, available at: http://www.w3.org/TR/xmlschema11-2/

Appendices

Appendix A Actors

An actor is considered to be a human resource that can do work in the context of a business process. Actors may be individual persons or groups of people acting together as an organization. As such, they can be identified by their name ‘attribute’ (e.g. the person identified as John Smith or the organization identified as Acme Construction) or by some combination of name and other identifying attributes where the unique identity of the actor is required.

Generally, within IDM, it is not the specific identity of an actor that is required. Rather, it is the role that an actor participating in an exchange requirement is required to play. That is, the concept of actor needs to be considered in terms of the roles that people or organizations play at a particular point in time. Any given ‘actor’ may simultaneously play more than one role. For this reason, the actor role should be seen as conceptually different to the actor.

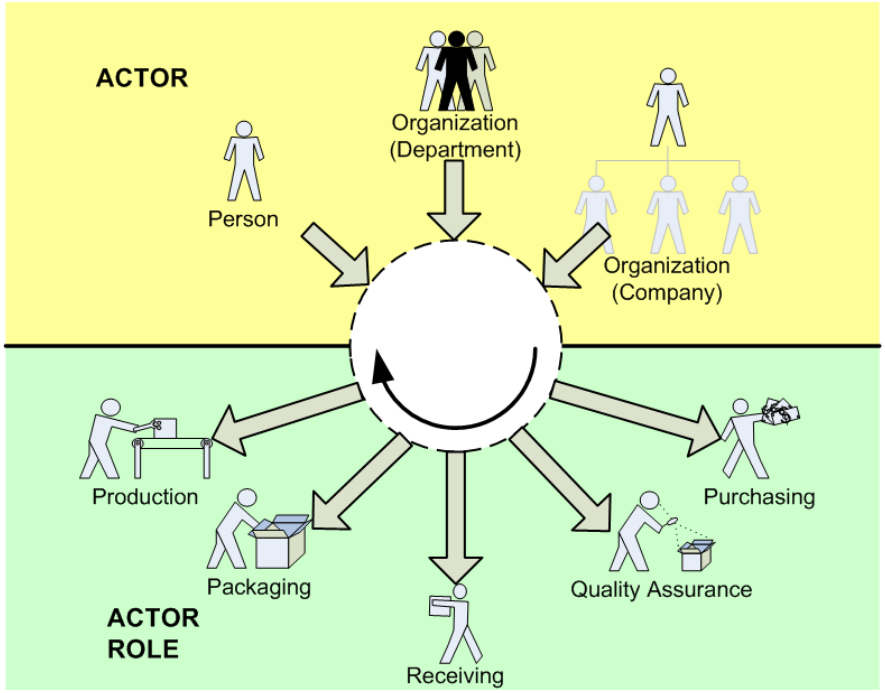


Figure 37: Actors and actor role

Roles themselves may have different purposes and any list of possible roles should ensure that that the purpose is met. For instance, we might see an actor as playing a professional role in respect of their day to day activity and a functional role in respect of the project. The professional role might be designated in conventional terms as e.g. ‘Architect’ whilst the functional role might be designated as e.g. ‘Building Design’.

Professional and functional roles should not be perceived as synonymous. Any actor may temporarily take on the role of ‘Building Design’ for a project according to circumstances. However, only a person suitably qualified may take on the professional role of ‘Architect’. This can be seen in an example of certification for building code compliance where any actor having the role of ‘Building Design’ might prepare a submission but only an actor professionally qualified as an ‘Architect’ can functionally take the role of ‘Qualified Person’.

Usage Roles

For certain, very general processes, the actors engaged in a business process will typically be identified by their role within that process. That is, the process itself implies a context for the role that an actor plays within its boundaries. This role may be different again to the functional and professional roles described above. For instance, in the context of a submission of a design for building code compliance, the roles may be ‘submitting agent’ and ‘licensed planning agent’ where a ‘licensed

planning agent' describes an actor with delegated building regulation approval powers. Here the role is in the context of the process concerned rather than being professional or functional.

Actor Role Specification

It is important to define a consistent list of roles that may be played by actors in a project to ensure consistency between exchange requirements.

The following list of roles is used in the development of the general set of exchange requirements. It has been derived from the Uniclass classification system. It is not complete but is intended to provide a starting point. The list identifies a primary role heading and a range of particular actor roles that may be required in the development of a construction project.

Contracting
- Contractors
- Sub-contractors

Design
- Architectural
- Building
- Interior
- Landscape

Engineering
- Acoustic
- Electrical
- HVAC
- Mechanical
- Public health
- Structural
- Road
- Telecommunications
- Water

Management
- Construction
- Design
- Project
- Facilities

Owner/Operator
- Client
- Lessee
- Operator
- Tenant

Planning
- Supervisors
- Town and country
- Transportation

Product supply
- Distributors
- Importers
- Manufacturers
- Suppliers

Statutory
- Building control
- Environmental health
- Fire prevention
- Health and safety

Surveying
- Building

Cost
Land
Quantity

External

Financial
Insurance
Leasing
Legal

Local Actor Roles

The list given above is used to provide 'standard' actor roles in the generic IDM exchange requirements. Where exchange requirements are extended for local use, a set of local actor roles should be defined. Local actor roles should be developed from published classification systems used by the construction industry within that geographic region.

When developing local actor roles, it is important to consider how information is to be dealt with and how precise you need to be in developing the exchange requirement. For instance, there is a general IDM actor role of 'Structural Engineering' but this could break down using Omniclass to:

- Structural Engineering
- Foundation Engineering
- High Rise Engineering
- Long-span Structure Engineering
- Tensile Structure Engineering
- Pneumatic Structure Engineering
- Hydraulic Structure Engineering

The fact that you can do something (such as having a fine grained actor role decomposition structure) does not necessarily mean that you should do it. For instance, it may be perfectly reasonable for IDM purposes to map the Omniclass 'Structural Engineering' to the Uniclass 'Structural Engineering'.

Unique Identification of Actor Roles

Actor roles used with the general IDM exchange requirements are captured within the international IFD dictionary.

It is strongly recommended that local actor roles that are to be used within IDM should also be captured with their own unique identity within the dictionary. In this way, mapping between actor roles expressed in local languages can be facilitated by using the dictionary identifiers and automatic conversion applied.

For instance, an electrical engineer is identified in typical Norwegian usage as Ingeniør Elektro'.

Actor Role Mapping as Business Rules

It is typically expected that the mapping between local and general IDM actor roles will be consistent. This means that 'global business rules' can be defined from the mappings such that, for example, every instance of general role 'A' becomes an instance of local role 'B'.

Where such a global approach is not relevant, business rules that are local to the exchange requirement can still be used.

Example of Local Actor Roles (Functional Roles)

The following table is taken from the Omniclass classification system. It is designated as 'Disciplines' within Omniclass but, from an IDM perspective, it acts as a table of 'Functional Roles'.

11 00 00 Planning

11 11 00 Regional Planning

- 11 21 00 Development Planning
- 11 31 00 Rural Planning
- 11 41 00 Urban Planning
- 11 44 00 Transportation
- 11 51 00 Environmental Planning
- 11 61 00 Facility Conservation Planning
  - 11 61 21 Historic Building Conservation Planning
  - 11 61 31 Ancient Monument Conservation Planning
  - 11 61 41 Archaeological Area Conservation Planning
- 11 99 00 Other Planning
  - 11 99 11 Master Planning
  - 11 99 14 Permitting
  - 11 99 17 Risk Assessment

21 00 00 Design

- 21 11 00 Architecture
  - 21 11 11 Residential Architecture
  - 21 11 21 Commercial Architecture
  - 21 11 24 Institutional Architecture
  - 21 11 27 Industrial Architecture
- 21 19 00 Drafting
- 21 21 00 Landscape Architecture
- 21 23 00 Interior Design
- 21 25 00 Specifying
- 21 27 00 Graphic Design
  - 21 27 11 Signage Graphic Design
- 21 31 00 Engineering
  - 21 31 11 Civil Engineering
    - 21 31 11 11 Geotechnical Engineering
  - 21 31 14 Structural Engineering
    - 21 31 14 11 Foundation Engineering
    - 21 31 14 21 High Rise Engineering
    - 21 31 14 31 Long-span Structure Engineering
    - 21 31 14 41 Tensile Structure Engineering
    - 21 31 14 51 Pneumatic Structure Engineering
    - 21 31 14 54 Hydraulic Structure Engineering
  - 21 31 17 Mechanical Engineering
    - 21 31 17 11 Plumbing Engineering
    - 21 31 17 21 Fire Protection Engineering
    - 21 31 17 31 Heating Ventilation and Air Conditioning Engineering
    - 21 31 17 33 Refrigeration Engineering
    - 21 31 17 34 Energy Monitoring and Controls Engineering
    - 21 31 17 37 Hydrological Engineering
  - 21 31 21 Electrical Engineering
    - 21 31 21 11 High Voltage Electrical Engineering
    - 21 31 21 21 Medium Voltage Electrical Engineering
    - 21 31 21 31 Low Voltage Electrical Engineering
  - 21 31 23 Communications Engineering
    - 21 31 23 11 Computer Network Engineering
    - 21 31 23 14 Alarm and Detection Engineering
    - 21 31 23 21 Audiovisual Engineering
  - 21 31 24 Process Engineering
    - 21 31 24 11 Piping Engineering
    - 21 31 24 21 Wind Engineering
    - 21 31 24 31 Co-Generation Engineering
    - 21 31 24 41 Nuclear Engineering
    - 21 31 24 51 Sanitary Engineering
  - 21 31 27 Military Engineering
  - 21 31 99 Other Engineering

- 21 31 99 11 Acoustical/Emanations Shielding Engineering
- 21 31 99 14 Antiterrorism/Physical Security Engineering
- 21 31 99 17 Value Engineering

21 99 00 Other Design
- 21 99 11 Fountain Design
- 21 99 14 Finish Hardware Design
- 21 99 15 Extraterrestrial Design Specialist
- 21 99 22 Health Services Design Specialist
  - 21 99 22 11 Hospital Design Specialist
  - 21 99 22 21 Nursing Home Design Specialist
- 21 99 24 Infrastructure Design Specialist
- 21 99 25 Irrigation Design Specialist
- 21 99 26 Laboratory Design Specialist
- 21 99 28 Lighting Design Specialist
- 21 99 29 Marina Design Specialist
- 21 99 31 Model Making Specialist
- 21 99 39 Solar Design Specialist
- 21 99 45 Transportation Design Specialist
  - 21 99 45 11 Air Transportation Design Specialist
  - 21 99 45 21 Roadway Transportation Design Specialist
  - 21 99 45 31 Marine Transportation Design Specialist
  - 21 99 45 41 Vertical Transportation Design Specialist
  - 21 99 45 51 Parking/Traffic Specialist Design Specialist

25 00 00 Project Management
- 25 11 00 Cost Estimation
- 25 14 00 Proposal Preparation
- 25 17 00 Client Briefing
- 25 21 00 Scheduling
- 25 31 00 Contract Administration
- 25 41 00 Procurement
  - 25 41 11 Manufacturing
    - 25 41 11 11 Building Product Manufacturing
    - 25 41 11 14 Process Manufacturing
    - 25 41 11 17 Construction Equipment Manufacturing
  - 25 41 14 Construction Product Sales
  - 25 41 17 Construction Product Marketing
  - 25 41 21 Construction Product Purchasing
- 25 51 00 Quality Assurance
- 25 54 00 Quality Control

31 00 00 Surveying
- 31 11 00 Building Surveying
- 31 21 00 Site Surveying
- 31 31 00 GIS (Geographical Information System) Engineering

41 00 00 Construction
- 41 11 00 Contracting
  - 41 11 11 General Contracting
  - 41 11 14 Subcontracting
    - 41 11 14 11 Masonry
    - 41 11 14 14 Carpentry
    - 41 11 14 17 Iron Working
    - 41 11 14 21 Plumbing Subcontracting
    - 41 11 14 24 Fire Protection Subcontracting
    - 41 11 14 27 Heating Ventilation and Air Conditioning Subcontracting
    - 41 11 14 28 Refrigeration Subcontracting
    - 41 11 14 37 Electrical Subcontracting
    - 41 11 14 51 Energy Monitoring and Controls Subcontracting
    - 41 11 14 54 Hydrological Subcontracting
    - 41 11 14 61 Painting

- 41 11 14 64 Tiling
- 41 11 14 66 Plaster Subcontracting
- 41 11 14 69 Gypsum Board Subcontracting
- 41 14 00 Construction Management

55 00 00 Facility Use Disciplines

- 55 14 00 Real Estate
  - 55 14 11 Real Estate Sales
  - 55 14 14 Property Appraising
  - 55 14 17 Leasing
- 55 21 00 Owner
- 55 24 00 Facility Operations
  - 55 24 11 Facility Space Planning
  - 55 24 14 Facility Management
  - 55 24 17 Facility Maintenance
  - 55 24 21 Facility Services
    - 55 24 21 11 Plumbing Operation and Maintenance
    - 55 24 21 14 Fire Protection Operation and Maintenance
    - 55 24 21 17 Heating Ventilation and Air Conditioning Operation and Maintenance
    - 55 24 21 21 Refrigeration Operation and Maintenance
    - 55 24 21 24 Energy Monitoring and Controls Operation and Maintenance
    - 55 24 21 27 Hydrological Operation and Maintenance
    - 55 24 21 31 Lightning Protection Operation and Maintenance
    - 55 24 21 34 Life Safety Operation and Maintenance
    - 55 24 21 37 Radiation Protection Operation and Maintenance
    - 55 24 21 41 Moisture Protection Operation and Maintenance
    - 55 24 21 44 Indoor Air Quality Evaluation
    - 55 24 21 47 Communications Operation and Maintenance
    - 55 24 21 47 11 Telecommunications Operation and Maintenance
    - 55 24 21 47 14 Information Technology Operation and Maintenance
  - 55 24 21 51 Facility Shielding Operation and Maintenance
    - 55 24 21 51 11 Acoustic Shielding Operation and Maintenance
- 55 36 00 Facility Restoration
  - 55 36 11 Concrete Restoration
  - 55 36 21 Masonry Restoration
  - 55 36 31 Parking Restoration
  - 55 36 41 Historic Restoration

81 00 00 Support Disciplines

- 81 11 00 Legal
  - 81 11 11 Code Specialist
  - 81 11 14 Forensic Specialist
- 81 13 00 Environment
  - 81 13 11 Environmental Impact
  - 81 13 14 Hazardous Materials Abatement
  - 81 13 44 Tree Preservation Specialist
- 81 31 00 Finance
  - 81 31 11 Banking
  - 81 31 14 Accounting
  - 81 31 17 Insurance
  - 81 31 21 Bonding
- 81 34 00 Human Resources

99 00 00 Other Disciplines

- 99 10 00 Science
- 99 13 00 Art
  - 99 13 11 Photography
- 99 40 00 Security
- 99 45 00 Public Relations
- 99 60 00 Education

Example of Local Actor Roles (Professional Roles)

The following table is taken from the Omniclass classification system. It is designated as 'Organizational Roles' within Omniclass. From an IDM perspective, it acts as a table of 'Professional Roles'.

11 00 00 Management
- 11 11 00 Executive Management
  - 11 11 11 Chief Executive
  - 11 11 21 Vice President
  - 11 11 31 Chairperson
  - 11 11 41 Board Member
  - 11 11 51 Partner
- 11 21 00 Middle-Management
  - 11 21 11 Supervisor
  - 11 21 21 Coordinator
  - 11 21 31 Trainer

21 00 00 Planning Roles
- 21 11 00 Developer
- 21 14 00 Owner
- 21 17 00 Planner
- 21 21 00 Cost Estimator
- 21 24 00 Scheduler
- 21 27 00 Contract Administrator

25 00 00 Design Roles
- 25 11 00 Space Designer
  - 25 11 11 Interior Designer
  - 25 11 14 Lighting Designer
  - 25 11 17 Space Planner
- 25 21 00 Architect
- 25 31 00 Engineer
- 25 41 00 Specifier
- 31 00 00 Procurement Roles
- 31 11 00 Manufacturer
- 31 21 00 Distributor
- 31 31 00 Product Representative
- 31 41 00 Buyer

35 00 00 Execution Roles
- 35 11 00 Surveyor
- 35 14 00 Contractor
- 35 17 00 Sub Contractor
- 35 21 00 Tradeperson
  - 35 21 11 Equipment Operator
  - 35 21 14 Laborer
    - 35 21 14 11 Skilled Laborer
      - 35 21 14 11 11 Master Craftsperson
      - 35 21 14 11 14 Journeyman
      - 35 21 14 11 17 Apprentice
  - 35 21 14 14 Unskilled Laborer
- 35 41 00 Inspector
  - 35 41 11 Code Inspector
  - 35 41 14 Safety Inspector

41 00 00 Utilization Roles
- 41 11 00 Facility Manager
- 41 21 00 Facility Maintenance
  - 41 21 11 Facility Engineer
  - 41 21 14 Maintenance Manager
- 41 31 00 Facility Services

- 41 31 11 Janitor
- 41 31 14 Window Washer

55 00 00 Support Roles
55 11 00 Administrative Support Staff
- 55 11 11 Administrative Assistant
- 55 11 14 Receptionist
- 55 11 17 Records Management Staff
- 55 11 21 Intern

55 14 00 Professional Support Staff
- 55 14 11 Consultant
- 55 14 14 Librarian
- 55 14 17 Draftsperson
- 55 14 21 Accountant
- 55 14 24 Lawyer

55 99 00 Other Support Roles
61 00 00 Groups
61 11 00 Team
61 21 00 Board
61 31 00 Committee
- 61 31 11 Task Team
- 61 31 21 Ad Hoc Committee

65 00 00 Organizations
65 11 00 Business Organizations
- 65 11 11 Corporation
- 65 11 14 Partnership
- 65 11 17 Sole Proprietorship
- 65 11 21 Joint Venture

65 21 00 Nonprofit Organizations
- 65 21 11 Association
- 65 21 14 Foundation
- 65 21 17 Union

Appendix B Project Stages

An exchange requirement is applicable to a business need at a particular stage in a project. Where exchange requirements are developed by different people in different places and at different points in time, an approach needs to be defined that can enforce consistency in the identification of the stages in a project. Without such consistency, it is not possible to bring exchange requirements together within an overall information framework. Effectively, lack of consistency in stage naming leads to a 'free for all' in which everyone can name project stages their own way. This is largely the case for older plans of work and conditions of engagement within the construction industry worldwide. However, to allow this within a digital information environment would lead to chaos.

The standard exchange requirements published in IDM are identified against project stages defined within the Generic Process Protocol (GPP). These are described below with their stage number as used in the exchange requirements documents, description and definition from the GPP. Stage 10 (disposal) is a modification to the standard GPP list to handle the final stage of a project lifecycle whilst stage 0 is a preliminary phase that is used to identify a preliminary requirement for a construction project and to determine its mission..

Stage	***Description***	***Definition***
Pre-project stages		
0	Portfolio requirements	Establish the need for a project to satisfy the clients business requirement
1	Conception of need	Identify potential solutions to the need and plan for feasibility
2	Outline feasibility	Examine the feasibility of options presented in phase 1 and decide which of these should be considered for substantive feasibility
3	Substantive feasibility	Gain financial approval
Pre-Construction stages		
4	Outline conceptual design	Identify major design elements based on the options presented
5	Full conceptual design	Conceptual design and all deliverables ready for detailed planning approval
6	Coordinated design (and procurement)	Fix all major design elements to allow the project to proceed. Gain full financial approval for the project
Construction stages		
7	Production Information	Finalise all major deliverables and proceed to construction.
8	Construction	Produce a product that satisfies all client requirements. Handover the building as planned.
Post-construction stages		
9	Operation and maintenance	Operate and maintain the product effectively and efficiently.
10	Disposal	Decommission, dismantle and dispose of the components of the project and the project itself according to environmental and health/safety rules

Table 1: Project stages adapted from the Generic Process Protocol

The stages within the Generic Process Protocol differ from those that are proposed within the international standard for project management (ISO 22263). This shows six stages which are:

- Inception
- Brief
- Design
- Production
- Maintenance
- Demolition

The mapping of these six stages to GPP stages can be seen in Figure 38 where ISO stages are shown in blue boxes whilst GPP stages are in red boxes

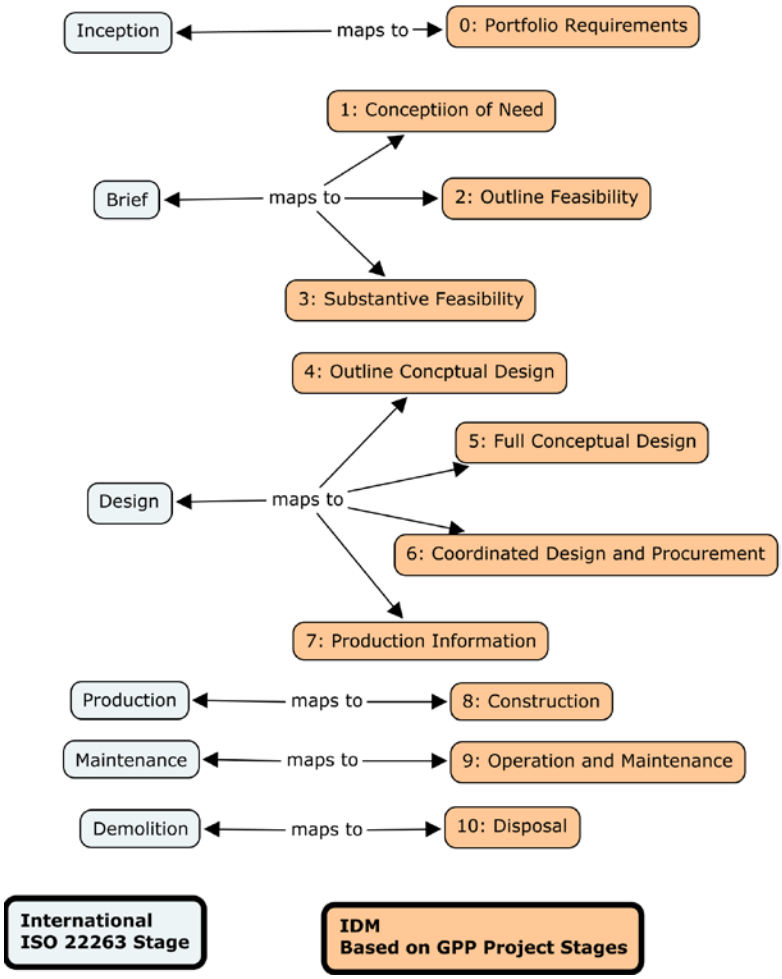


Figure 38: Mapping between ISO and GPP stages

Local Project Stages

Project stages are usually defined according to local process protocols. That is, the identification of project stages in one place will differ from the identification of project stages in another place. For example, project development is often organized according to the RIBA Plan of Work within the UK and according to the HOAI protocol in Germany.

The project stages within a standard exchange requirement can be customised to reflect local practice within a local exchange requirement. That is, the standard table can be replaced by a locally defined table. Local exchange requirements can be defined according to this local protocol.

Where local protocols are used, it is a good idea to ensure that there is a mapping between the stages in the local protocol and the standard stages. Either:

- a single standard stage is decomposed into multiple stages in the local protocol
- multiple standard stages are composed into a single stage in the local protocol

Note that standard stages and local protocol stages should always conform to boundaries such that there is a 1:1, 1:many or many:1 relationship between them. Project stages must not cross boundaries such that a stage in a local protocol starts part way through one standard stage and ends part way through another standard stage.

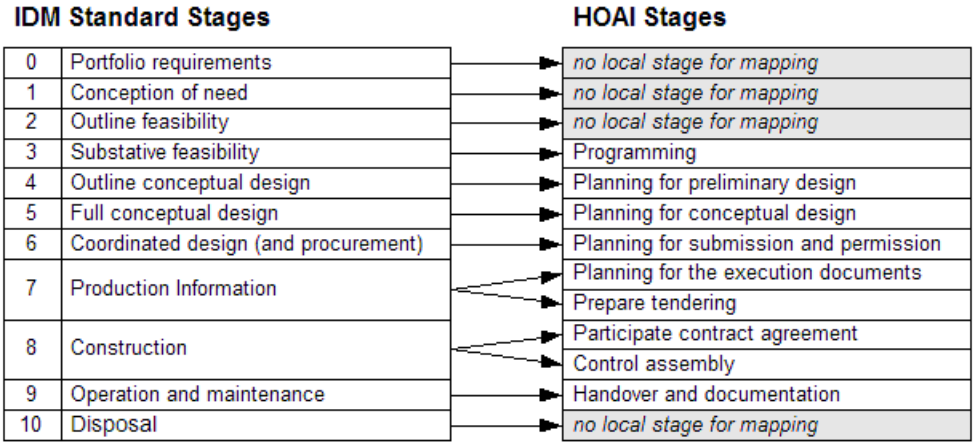


Figure 39: Mapping HOAI protocol stages (Germany) to IDM standard stages

In the example above showing a mapping between the German HOAI protocol and the IDM standard stages, some standard stages are decomposed to greater detail in HOAI. Similarly, some standard stages do not have a direct equivalent in HOAI. In this case, either the exchange requirement is not relevant to the local protocol (unlikely) or the standard stages all map to the closest HOAI stage. Thus standard stages 0, 1, 2 and 3 would map to the HOAI stage 'Programming'.

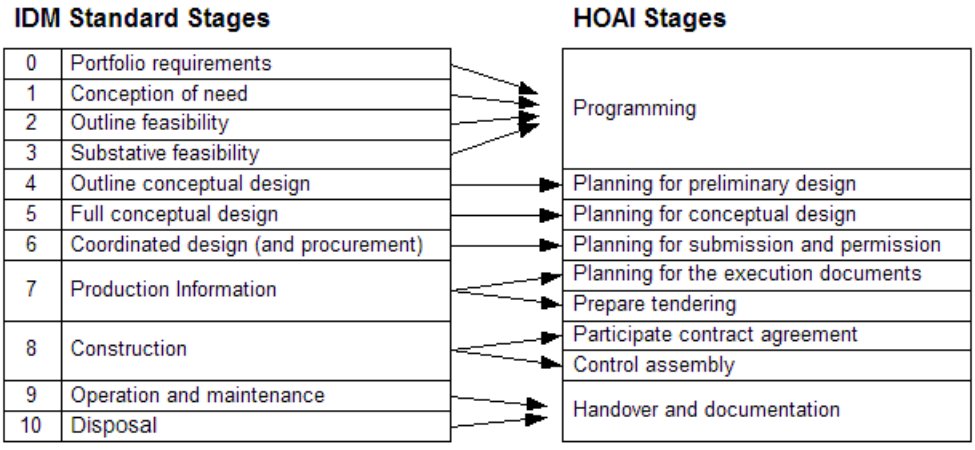


Figure 40: Mapping HOAI protocol stages with empty stages removed s

If we consider the HOAI stages being numbered as for the standard stages and we set the number for 'Participate contract agreement' as 8a and 'Control assembly' as 8b then the standard stage mappings are:

IDM (8) ==> Local (8a)
IDM (8) ==> Local (8b)

Practically, this would mean that an exchange requirement relevant to standard stage 8 would need to be either:

- Identified as a local exchange requirement at either local stage 8a or 8b
- Separated into two separate local exchange requirements covering local stages 8a and 8b

Multiple Project Stage Mapping

If we now consider the ISO, GPP and HOAI project stages from the above, it is clear that they can all be brought together into a single map, viz:

Example of Local Project Stages

As with the local identification of actors, local identification of project stages can frequently be found in classification system tables. The following example is from the Omniclass table 31.

The taxonomy elaborated in this table appears to more closely define a list of activities (characterized as phases) under general stage headings. Therefore, care needs to be exercised when using such a table to ensure that the depth of project stage identification does not become unreasonable and unworkable. Many of the items within this table themselves appear to be candidates for identification of exchange requirements rather than being candidates for identification of project stages.

10 00 00 Conception Stage
- 10 11 00 Needs Identification Phase
  - 10 11 14 Project Description Phase
  - 10 11 17 Imagination Phase
  - 10 11 21 Inception Phase
- 10 14 00 Concept Phase
  - 10 14 14 Market Analysis Phase
  - 10 14 15 Research Phase
  - 10 14 17 Proect Feasibility Phase
  - 10 14 21 Project Programming Phase
  - 10 14 24 Project Planning Phase
- 10 27 00 Pre-Design Phase
- 10 31 00 Budgetary Estimate Phase
- 10 34 00 Definition Phase
- 10 37 00 Conceptual Phase
  - 10 37 11 Data Collection Phase
  - 10 37 14 Site Survey Phase
  - 10 37 21 Geotechnical Investigation Phase
- 10 41 00 Preliminary Design Phase
  - 10 41 11 Schematic Design Phase
  - 10 41 21 Design Development Phase
  - 10 41 31 Preliminary Estimate Phase
  - 10 41 34 Technical Studies Phase
  - 10 41 37 Public Consultation Phase
  - 10 41 41 Analysis Phase
  - 10 41 47 Environmental Studies Phase
  - 10 41 51 Due Diligence Phase
  - 10 41 54 Site Selection Phase

10 41 57 Property Acquisition Phase
10 41 61 Planning Phase
10 43 00 Scheduling Phase
10 45 00 Budgetting Phase
15 00 00 Project Delivery Selection Stage
15 10 00 Project Delivery Evaluation Method Phase
15 20 00 Team Assembly Phase
15 20 11 Design Team Selection Phase
15 20 14 Project Team Selection Phase
15 20 17 Design/Builder Selection Phase
15 20 21 Construction Manager Selection Phase
15 20 24 Request for Qualifications (RFQ) Phase
15 20 27 Request for Qualifications (RFQ) Evaluation Phase
15 20 31 Request for Proposals (RFP) Phase
15 20 34 Request for Proposals (RFQ) Evaluation Phase
15 20 37 Interview Phase
20 00 00 Design Stage
20 10 00 Preliminary Project Description Phase
20 10 11 Preliminary Engineering Phase
20 10 14 Conceptual Design Phase
20 10 17 Schematic Design Phase
20 10 21 Preliminary Design Phase
20 20 00 Design Development Phase
20 20 11 Detailed Design Phase
20 20 14 Final Design Phase
20 20 17 Prototype Design and Testing Phase
20 20 21 Engineering Analysis Test Phase
20 20 24 Product Selection Phase
20 20 27 Material Selection Phase
20 20 31 Equipment Selection Phase
20 20 34 Estimating Phase
20 20 37 Value Analysis Phase
25 00 00 Construction Documents Stage
25 10 00 Construction Documents Preparation Phase
25 10 11 Construction Data Preparation Phase
25 10 14 Drawing Preparation Phase
25 10 17 Detail Preparation Phase
25 10 21 Fabrication Drawing Preparation Phase
25 10 24 Coordination Drawing Preparation Phase
25 10 27 Specifications Preparation Phase
25 10 31 Project Manual Preparation Phase
25 20 00 Construction Documents Production Phase
25 30 00 Construction Cost Estimating Phase
30 00 00 Procurement Stage
30 10 00 Solicitation Phase
30 10 11 Advertising Phase
30 20 00 Pre-Qualification Phase
30 30 00 Bidding Phase
30 30 11 Bid Scoping Phase
30 30 14 Pricing Phase
30 30 17 Bid Preparation Phase
30 30 21 Proposal Preparation Phase
30 40 00 Selection Phase
30 40 11 Contractor Selection Phase
30 40 14 Proposal Evaluation Phase
30 40 17 Bid Evaluation Phase
30 50 00 Contracting Phase
30 50 11 Negotiation Phase

30 50 14 Contract Award Phase
30 50 17 Contract Execution Phase
40 00 00 Execution Stage
40 10 00 Construction Contract Administration Phase
40 20 00 Construction Preparation Phase
40 20 11 Mobilization Phase
40 20 14 Subcontracting Phase
40 20 17 Permitting Phase
40 20 21 Regulatory Review Phase
40 20 24 Regulatory Approval Phase
40 20 27 Submittal Processing Phase
40 20 31 Scheduling Phase
40 20 34 Coordination Phase
40 20 37 Buy-Out Phase
40 30 00 Product Development Phase
40 30 11 Product Prototyping Phase
40 30 14 Product Testing Phase
40 30 17 Product Evaluation Phase
40 30 21 Product Purchasing Phase
40 30 24 Product Ordering Phase
40 30 27 Product Delivery Phase
40 40 00 Construction Phase
40 40 11 Construction Start-Up Phase
40 40 11 11 Construction Process Validation Phase
40 40 11 14 Mobilization Phase
40 40 11 17 Installation Phase
40 40 11 21 Application Phase
40 40 11 24 Production Phase
40 40 11 27 Tooling and Equipment Phase
40 40 14 Project Execution Phase
40 40 14 11 Erection Phase
40 40 14 14 Assembly Phase
40 40 14 17 Manufacturing Phase
40 40 14 21 Off-Site Fabrication Phase
40 40 14 24 Fabrication Phase
40 40 14 27 Demobilization Phase
40 40 91 Close of Construction Phase
40 40 91 11 Facility Testing Phase
40 40 91 14 Adjusting and Balancing Phase
40 40 91 17 Evaluation Phase
40 40 91 21 Cleaning Phase
40 40 91 24 Contract Closeout Phase
40 40 91 27 Out-Fitting Phase
40 40 91 31 Turnover Phase
40 50 00 Commissioning Phase
40 50 11 Commissioning Start-Up Phase
40 50 21 Commissioning Monitoring Phase
40 50 31 Commissioning Certification Phase
40 50 41 Facility Equipment Demonstration Phase
40 50 51 Facility Equipment Training Phase
50 00 00 Utilization Stage
50 10 00 Occupancy Phase
50 10 11 Pre-Opening Phase
50 10 14 Grand Opening Phase
50 10 17 Use Phase
Deployment Phase
50 20 00 Facility Management Phase
50 20 11 Facility Operation Phase

50 20 21 Facility Inspection Phase
50 20 31 Facility Maintenance Phase
50 30 00 Facility Re-Use Phase
50 30 11 Facility Renovation Phase
50 30 14 Facility Remodeling Phase
50 30 17 Facility Expansion Phase
50 30 21 Facility Restoration Phase
50 70 00 Facility Repair Phase
60 00 00 Closure Stage
60 10 00 Adaptation Phase
60 10 11 Re-Fitting Phase
60 10 14 Re-Commissioning Phase
60 20 00 Redevelopment Phase
60 25 00 Relocation Phase
60 30 00 Deconstruction Phase
60 35 00 Recycling Phase
60 40 00 Decommissioning Phase
60 45 00 Demolition Phase

Appendix C IDM Naming Rules

The IDM methodology proposes a set of rules for naming exchange requirements and functional parts. These rules are developed from basic English language grammar with IDM specific extensions.

The aim of the naming rules is to define a restricted number of allowed action types (verbs). In particular, actions associated with IFC relationships are predefined by their relationship type e.g. associate, define, connect, assign etc.

This will create a dictionary of actions that can be used in the representation of processes and thus, in combination with IFC entities, extend the common language used on AEC/FM projects.

General

- All identifiable words in a name are separated using an underscore character '_'
- All characters used in naming IDM elements shall be in lower case

Process Maps

- Each process map has a name that consists of two parts
  - The first part is the prefix 'pm' which identifies that this component is an process map
  - The second part of the name is the subject of the process map and is expressed as a noun or noun phrase e.g. cost_model, energy_analysis etc.

Exchange Requirements

- Each exchange requirement has a name that consists of three parts
  - The first part is the prefix 'er' which identifies that this component is an exchange requirement
  - The second part of the name is an action (or activity) required and is expressed as a verb
    - All exchange requirements have the action 'exchange'; thus the first part of the name of an exchange requirement will always be 'er_exchange_'
  - The third part of the name is the subject of the exchange requirement and is expressed as a noun or noun phrase e.g. space_model
- Exchange requirements may have parameters that enable further qualification. Parameters are expresses as a list within parentheses ( ) or [ ] or { }
  - Parameters for exchange requirements typically restrict the exchange requirement by project stage and/or actor role e.g. er_exchange_space_model (outline)

Exchange Requirement Models

- Each exchange requirement model has a name that reflects the underlying exchange requirements except that:
  - The first part is the prefix 'erm' which identifies that this component is an exchange requirement model

Functional Parts

- Each functional part has a name that consists of three parts
  - The first part is the prefix 'fp' which identifies that this component is a functional part
  - The second part of the name is an action (or activity) required and is expressed as a verb. Various actions can be applied from an allowed range that includes:
    - Model – key ideas that are expressed
    - Define – provision of a set of properties (a property set within IFC)
    - Represent – a form of geometric representation
    - Select – alternative selections that may be available
    - Set – values that may be set for a purpose
    - Present – an image, including annotation elements used to embellish the image

- The third part of the name is an object that receives the action and is expressed as a noun or noun phrase). This may be a direct object (as in 'model wall') or an implied indirect object (as in 'associate material' which means associate {to wall} {the} material)

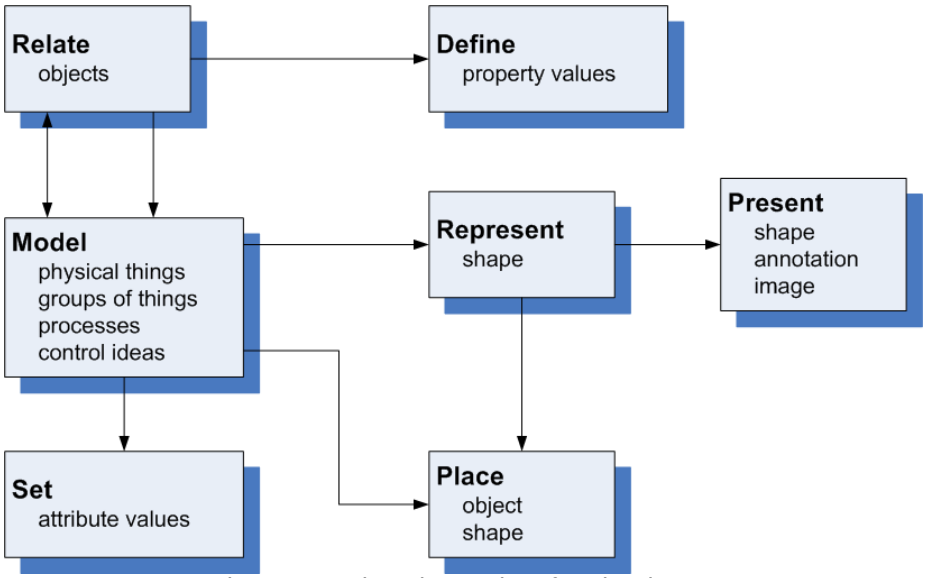


Figure 41: Actions that apply to functional parts

- Functional parts that handle IFC relationships are named according to the relationship in the IFC model (since this already represents a verb phrase). Typically, the IfcRel prefix is stripped from the relationship name and the remaing words separated by the underscore character. For instance, fp_associates_classification handles the association of a classification to an object as provided by the entity IfcRelAssociatesClassification.
- Functional parts may have parameters that enable further qualification. Parameters are expresses as a list within parentheses ( ) or [ ] or { }. Allowed parameters are determined by the context of the functional part and are described within the function al part documentation.

Business Rules

- Each set of business rules has a name that consists of three parts
  - The first part is the prefix 'br' which identifies that this component is a business rules set
  - The second part of the name is a location identifier that may be used to specify the place or project at which the business rules set applies. For an internationally generic rule set, the location should be identified as 'GENERIC'.

    More specific rules in location identification may be applied at a particular place. Location identifications may be cascaded to provide for progressive localization of an exchange requirement.

  - The third part of the name is the numeric sequence identifier for the business rule set.
  - The fourth and final part of the name is the exchange requirement to which the business rule set applies (excluding the erm_ prefix)

Concepts

- Rules for naming and identifying concepts are as given in the Model View Definition methodology

Appendix D IDM Use of BPMN Methods

In developing process maps, IDM uses some techniques of the Business Process Modeling Notation (BPMN) in particular ways. These techniques apply some minor restrictions to the overall development. However, all of the techniques are legal uses of the notation.

- Each process map is developed as a single 'pool'. This is intended to identify each project as an overall organization and each actor participating as a part of that organization.
- The name of the pool identifies the process map
- Activities for each actor are shown in individual swimlanes
- The 'Information Model' is shown as a separate actor role within its own swimlane.
- An 'Exchange Requirement' is always shown in a process map as a data object within the Information Model swimlane.
- The provision of an 'Exchange Requirement' is always shown by:
    - The definition of an event within the 'sequence flow' of the process map
    - The designation of the event as a message
    - A message flow from the event to the Exchange Requirement data object
    - Specification of the Exchange Requirement as a particularly identified data object within the Information Model swimlane
- Where an Exchange Requirement is used as an input to the process map, this is shown exactly as above except that the message flow is from the data object to the event.
- Where an Exchange Requirement export event is defined, it is succeeded immediately by a gateway. This is referred to as a 'Coordination Gateway'.
- A Coordination Gateway marks a point in the process at which the information from several exchange requirements may be brought together for examination, consideration or action.
- A Coordination Gateway is an exclusive (XOR) data driven gateway i.e. it brings together and coordinates data based on which decisions can be made.
- Each process within a process map shall have an identifier. Typically, this shall be a number with each level in the process map separated by a point (e.g. 4, 4.1, 4.1.3 etc.)

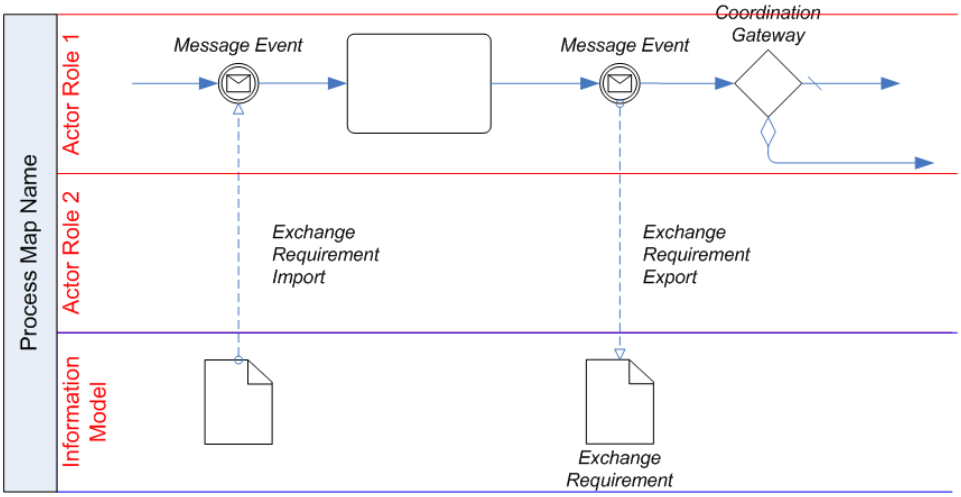


Figure 42: BPMN elaborations used in IDM

Appendix E IDM Publishing Guide

What is Confluence

This section describes the publication of IDM components to the web using the Confluence software application.

Confluence allows direct entry and editing of page content to the world wide web. Such applications are termed Wiki's. They allow for rapid development of content and linking of content between different pages to provide a knowledge resource.

Since the complete content of IDM represents a potentially huge knowledge resource for the building construction industry, the use of a Wiki tool to facilitate and manage the content was considered appropriate.

Content

Content is organized as a set of pages. Each page is created directly in Confluence. Figure 43 shows a Confluence edit screen and the various components.

Edit Guide

The content of a Confluence page can be formatted in various ways. Typically format instructions are shown in the 'Edit Guide' to the right of the screen when editing.

This does not show the entire set of format and edit notations within Confluence. A complete guide can be obtained through the full notation guide link located at the bottom of the Edit Guide.

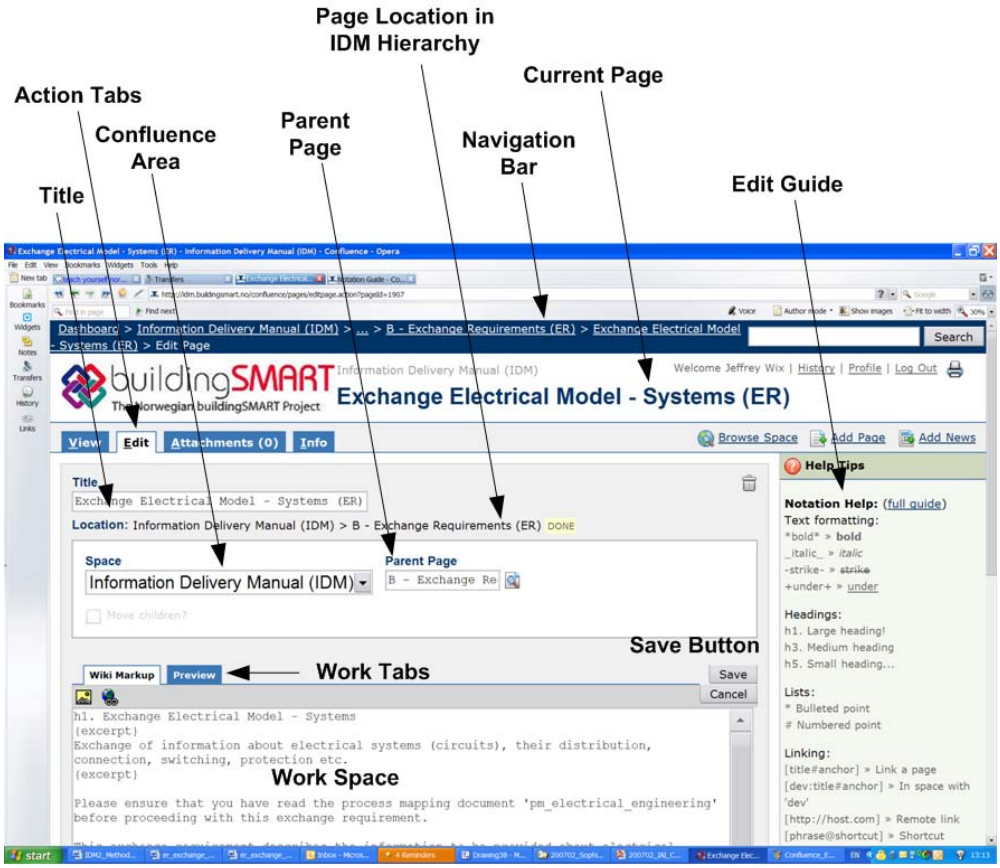


Figure 43: Edit screen in Confluence

Action Tabs

The current action is determined by the Action Tab that has been selected.

'View' allows you to see content as it has been entered without being able to change it.
'Edit' (as shown) allows content to be changed
'Attachments' allows files (particularly graphic images) to be attached to a page.
'Info' provides general information about activities on the present 'Confluence Area

Title

Documents in IDM may comprise several pages. Each page has a title. This is an identifying label that is used to link to the page from other pages. In a hierarchy of child pages, it is the title that is used to identify the content of each page.

Location

The location of each page is identified by reference to its parent page. Note that pages can be moved from one parent to another by editing the location. However, it is recommended that pages should be created in the location at which they will be required and should not be moved except by a member of the IDM Technical Team.

Example

Location: Information Delivery Manual (IDM) > B - Exchange Requirements (ER)

The location of a page can be changed by pressing the EDIT marker adjacent to the Location. This opens the window showing the parent page below it. To change the location of a page, simply enter the name of the parent page to which it should belong.

Parent Page

Confluence enables a hierarchical structure of pages to be defined where each page belongs to a parent. This allows the identity of the immediate parent page to be seen and, if necessary, change (see 'location' above).

Confluence Area

This identifies the particular Confluence area in which you are working (in case several areas are defined).

Current Page

The title of the page currently being viewed or edited is displayed at the top of the Confluence screen.

Navigation Bar

The navigation bar provides a full description of the IDM hierarch leading up to the current page. Note that if the text within the hierarchy is too long to display, it is broken in the middle with an ellipsis marker (…).

Each element within the navigation bar is active. You can navigate directly to a level shown by clicking on it. If you click on an ellipsis marker (because the page you want to navigate to is not presently shown), the display changes to show the pages at that point.

Work Space

The lower part of the screen shows the Confluence work space. This is where you enter and format or view text and graphics.

Work Tabs

The method of working (in Edit mode) can be selected using the work tabs. Normal editing is done through selecting the 'Wiki Markup' tab. The results of working within the Wiki Markup area can be previewed without leaving Edit mode by selecting the 'Preview' tab.

Save Button

The results of editing pages in Wiki mark-up can be saved and viewed by clicking on the 'Save' button.

IDM in Confluence

In this section, some of the specific points with regard to the publication of IDM content in Confluence are presented.

Entry to IDM

Entry to the IDM area of Confluence is via the 'dashboard' (where all areas/spaces available are visible). The dashboard is the first screen seen on entry to Confluence.

Select 'Information Delivery Manual' from the Spaces: All' part of the dashboard.

This brings you to the main IDM page. Here, an introduction to IDM is given. You can also see all of the current IDM content which includes process maps, exchange requirements, functional parts and other content.

You can navigate to any IDM component see in the Contents list by clicking on it.

You can also see groups of 'children' pages at the bottom of the page. By following the IDM's child, you can see the tree structure (hierarchy) of IDM information organization.

Creating a New IDM Component

An IDM component is a process map, exchange requirement or functional part. A new component requires the creation of a head page that acts as the 'root' for the component in the hierarchy

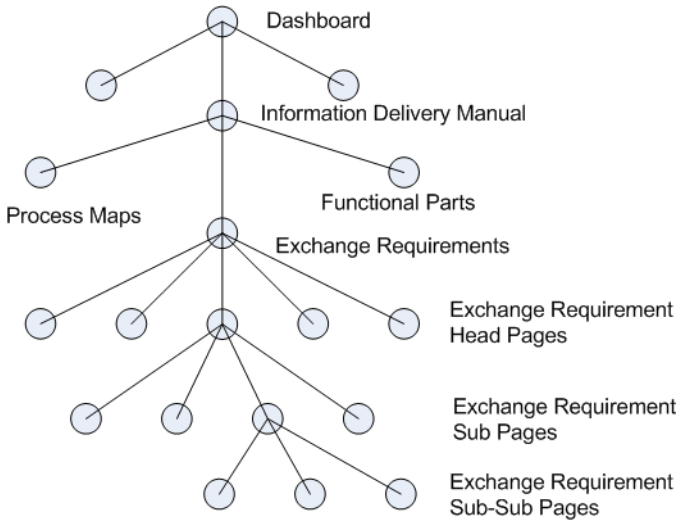


Figure 44: IDM tree structure

There are various ways to create a new page in IDM. Note that, as with any Wiki site, any reference to a page may cause that page to exist. That is, the page doesn't have to be created before you link to it. The act of creating the link provides the opportunity for the creation of the page. This may sound a little complicated; it should become clearer as we go on.

The simplest way to create a page is through the 'child' section of a page. If a page already has a child, this will be shown. To create a new child page, simply click on 'Add Child Page'

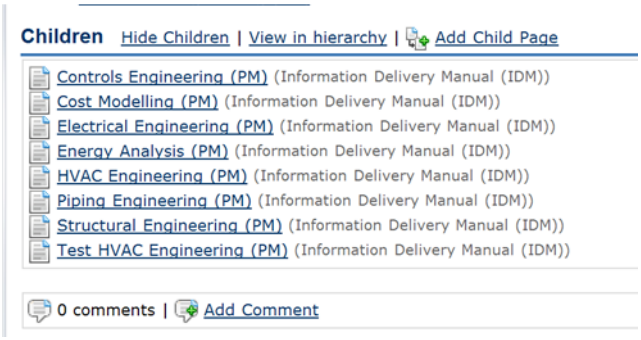


Figure 45: Children pages display

New pages may be created by reference. This is done by entering a link word and a page title as follows:

```
[Precursor|0 - Precursor - Exchange Electrical Model - Systems (ER)]
```

The text is within square brackets. A link term and a title are separated by a pipe (|) character. When saved and viewed, this appears initially as Precursor+. The '+' character indicates that the child page has not yet been created. This can be done by clicking on the link word. This creates a page with the title given.

Templates

New pages may also be created from templates. A template is simply a predefined IDM document that contains all the headings and required layout together with some basic instructions for creating the IDM part. Templates are available at this time for:

- Process maps
- Exchange Requirements
- Functional Parts

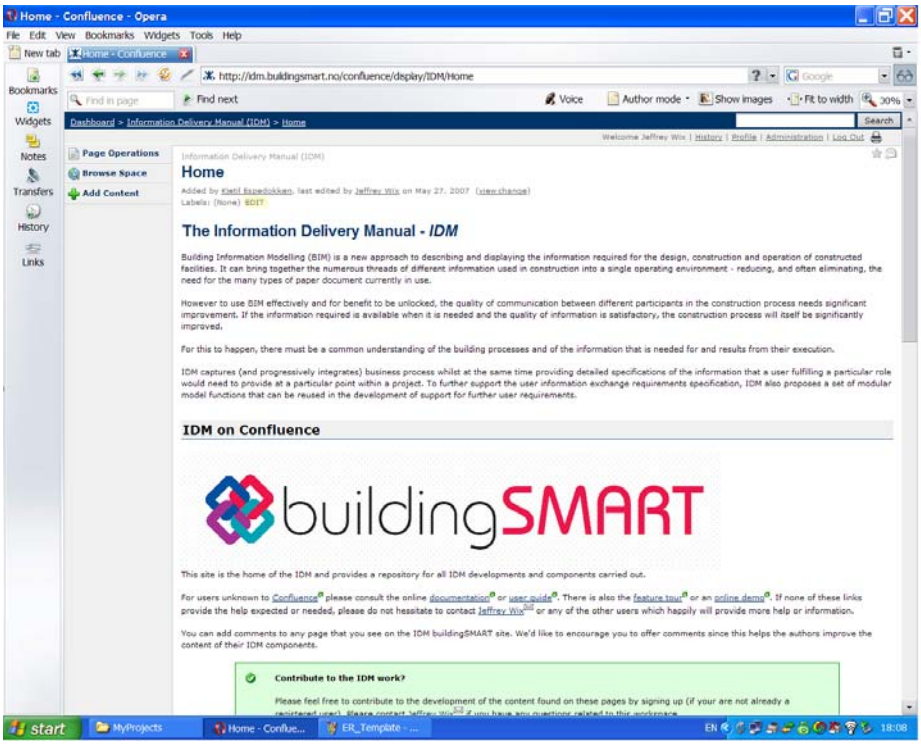


Figure 46: IDM in Confluence home page

To create a template, go to the home page for IDM in Confluence.

In the panel to the left of the home page, select ‘Browse Space’.

A sub menu will provide a set of choices from which ‘Template’ should be selected.

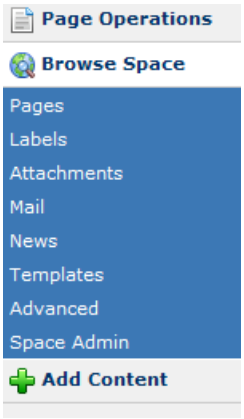


Figure 47: Using template from the ‘Browse Space’ menu

The available templates will now be displayed. Adjacent to each is the option to create a page from a template. This should be selected. The required template will now be presented.

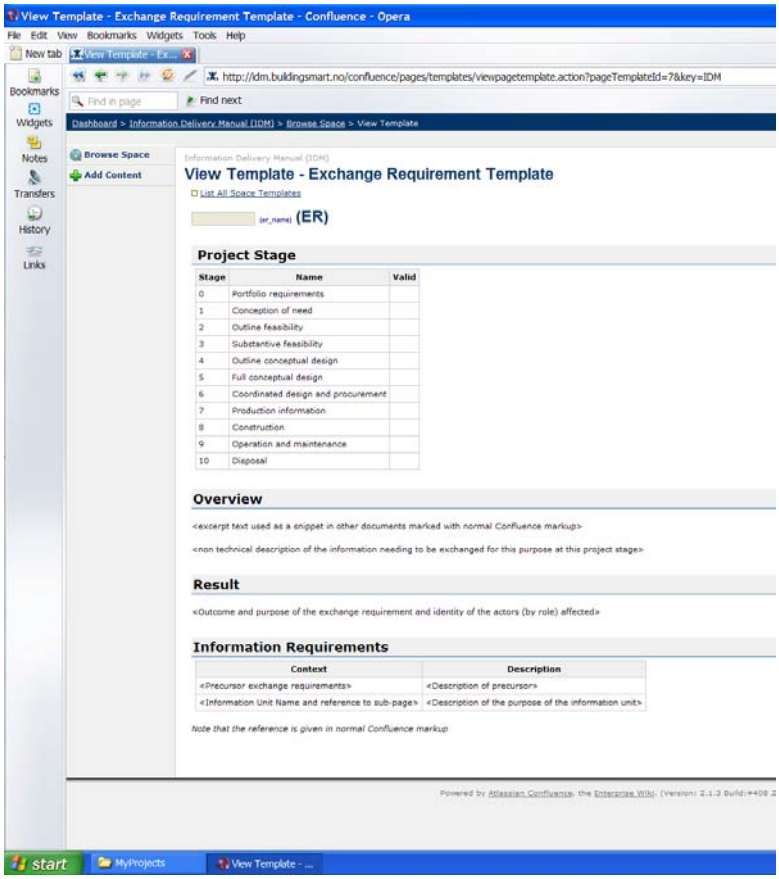


Figure 48: The Exchange Requirement template

Each template has predefined questions that allow the name and relative location of the completed document in the IDM hierarchy to be defined. Once these questions have been answered, the content for the IDM document can be completed as for any other IDM document. Once the document is completed, it can also be edited in the same way as any other IDM document.

Adding a Page Directly

IDM pages can also be added directly from the 'Add Content' option available at the IDM home page by selecting 'Add Page'. This will allow the user to define a page that does not initially have a location in the IDM document hierarchy. The location will need to be provided directly by the user through the editing process.

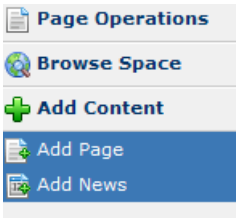


Figure 49: Adding a page directly from the 'Add Content' sub menu

Excerpts

An excerpt is a marked piece of text that can be referenced by and consequently used within another page. It is usually located at the beginning of a page.

Excerpts are particularly useful to show the nature of the content of one page within another page (that references the excerpt). They can therefore be very helpful in ensuring consistency between textual elements on all pages.

```
{excerpt}
Some text that will be displayed as a small part of the total
{excerpt}
```

Appendix F IDM EXPRESS Creation Guide

Notation

A primary notation used to define schemas for functional parts is EXPRESS-G as set out in ISO 10303 Part 11. For IDM functional parts however, an extension to the standard notation is used to identify other functional parts that are referenced and that therefore contribute additional entities to the complete schema.

The extension to the notation identifies that the functional part being referenced acts as a placeholder for a part of the schema that is not fully elaborated within the functional part currently being defined. This is required since the conventional USE FROM and REFERENCE FROM 'entity inclusion' capabilities within the formal EXPRESS-G specification do not fully satisfy the requirements of IDM.

The form of the extension is that an entity is defined in which the name of the functional part is inserted according to the IDM naming rules (i.e. in lower case text with underscore characters separating each significant term).

In documentation for IDM, the Visual EXPRESS editor from EPM Technology is extensively used. This provides EXPRESS-G diagrams in color. Default color for entities is yellow (although this can be varied). When creating documentation, it is recommended that the entity describing the functional part placeholder is shown graphically in a separate color. The actual color used is not significant; it's purpose is simply to act as a visual clue.

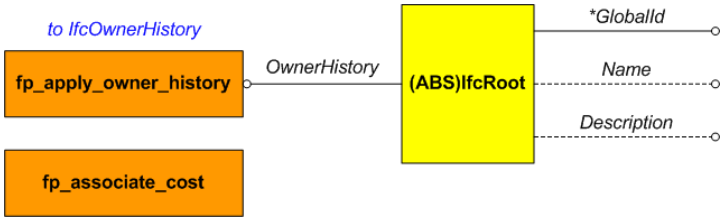


Figure 50: IDM functional part placeholder in EXPRESS-G

The diagram shows an example of how referenced functional parts are shown in EXPRESS-G diagrams within IDM.

A functional part may include an entity that is directly attributed (from another entity). This should be shown by showing the attribute as normal but making the functional part as its target. A label should also be added to the functional part entity showing the entity within it to which the attribute is associated.

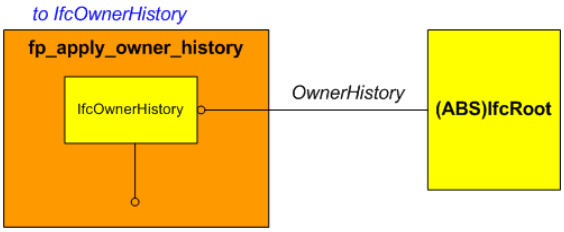


Figure 51: Decomposing functional part attribute

Note that where the functional part deals with a relationship, this is shown without attribution to an entity (since this is handled within the definition of that functional part).

Note also that defined datatypes (such as IfcLabel, IfcText etc.) are shown in abbreviated form. These datatypes are fully elaborated in the EXPRESS schema accompanying the EXPRESS-G diagram.

~~Appendix G – IDM Issues Log~~

~~An 'Issues Log' is maintained for IDM. Access to the issues log is via the IDM web site at http://idm.buildingsmart.com.~~

~~Issues may be entered by anyone observing or using any of the IDM components. Entry of the issue is via a configured version of the JIRA tracking software. This is the same software that is used by the buildingSMART® Model Support Group for tracking IFC model issues and is similarly maintained and monitored.~~

~~For each IDM issue entered, it is important that the following rules are observed:~~

- ~~The type of IDM component concerned must be identified (process map, exchange requirement, functional part, business rules).~~
- ~~The name of the component must be given~~
- ~~The revision of the component to which the issue relates must be quoted.~~
- ~~The issue must be clearly stated in technical terms. Issues that are not described adequately or that do not have a technical foundation will be rejected by default.~~
- ~~A proposal must be made for a solution. Issues that do not propose a solution (however simple) will be rejected by default.~~
- ~~Access to the IDM issues log can also be obtained from other buildingSMART® technical web resources that use the JIRA issues tracking software.~~

IDM and IFC Issues

During the course of working on or reviewing IDM components, it is quite possible that issues concerning the IFC schema may become apparent. These should be entered directly through the IFC Issues Resolution Database.