An Integrated Process for Delivering
IFC Based Data Exchange

Authors
Richard See – buildingSMART International MVD Coordinator, Co-Chair buildingSMART
Technical Committee, Chairman – The BLIS Consortium
Jan Karlshoej – buildingSMART International IDM Coordinator
Diane Davis – buildingSMART alliance IDM Coordinator

Date
09-Sep-11

Version
1.0

Status
Proposal to both buildingSMART and The BLIS Consortium

History
01-Jul-11 – Draft 1 for Version 1
*Note: this document incorporates material defining the MVD process and deliverable formats developed by Jiri Hietanen and adopted as the standard for MVD by bSI
25-Jul-11 – Draft 4: added several previous works – mostly into the MVD sections
18-Aug-11 – Draft 4.2: several improvements to IDM sections
26-Aug-11 – Draft 5 – major edit of Exec Summary, Acknowledgments, Background, Overview, MVD, Certification, and BIM Validation sections
05-Sep-11 – Draft 6 – major edit of MVD sections

Copyright
Copyright © 2011 buildingSMART International and The BLIS Consortium

NOTE: This is a draft document under development – some parts can be missing of incomplete.
Its intention is only to present some suggestion for revision of Part 1 and development of Part 3 of the ISO 29481 series, note state final concepts and formulations.

Table of contents
Executive summary ....................................................................................................................................... 3
Background and Acknowledgements ........................................................................................................... 4
An Integrated Process .................................................................................................................................. 5
Process Overview ..................................................................................................................................... 5
Coordination of IDM/MVD Projects ........................................................................................................... 8
Organizational Endorsement .................................................................................................................... 9
Developing an IDM ..................................................................................................................................... 10
Experience Required for IDM Development ........................................................................................... 10
IDM Coordination .................................................................................................................................... 10
Forming a Domain Group ....................................................................................................................... 10
IDM Deliverables .................................................................................................................................... 10
Reusable Exchange Concepts ............................................................................................................... 15
IDM Coordination .................................................................................................................................. 16
Developing an MVD .................................................................................................................................... 22
MVD Background ................................................................................................................................... 22
The context for MVD ............................................................................................................................... 23
Goals for MVD ........................................................................................................................................ 24
Reusable MVD Concepts ....................................................................................................................... 24
Experience Required for MVD Development ......................................................................................... 25
MVD Coordination ................................................................................................................................... 25
MVD Deliverables ................................................................................................................................... 25
Executive summary

For many years, the International Alliance for Interoperability (IAI), now buildingSMART\(^1\), struggled to deliver reliable exchange of Building Information Model (BIM) data between stakeholders in building construction projects. The Industry Foundation Classes (IFC)\(^2\), a comprehensive information model schema for the building industry, was developed as to be an open industry standard for such exchanges. The IFC model is necessarily large and complex, as it includes all common concepts used in building industry projects, from feasibility analysis, through design, construction, and operation of a built facility. Implementing support for all of IFC is therefore beyond the scope of any one application, with the possible exception of model servers\(^3\). For the first few releases of IFC, software vendors struggled to deliver software that would deliver, and buildingSMART struggled to develop certification testing that would ensure, the target ‘reliable BIM exchange.’

Process – This document defines an integrated process for designing, implementing, certifying and using standard information exchanges for the global building industry. This process includes 4 distinct phases that begin with requirements definition and end with use of the information exchange in building industry projects.

For buildingSMART and BLIS, the information model schema currently used to coordinate and make various information exchanges in the Industry Foundation Classes (IFC)\(^4\) and the source for coordinated terminology is the International Framework for Dictionaries (IFD)\(^5\).

![Figure 1: Transformation of needs into operational solutions](image)

Phase 1: IDM – an Information Delivery Manual defines an industry process that involves at least two types of software applications, and the information that should be exchanged between those applications. IDMs include four primary deliverables, using standard formats. These are: Process Maps (which define the industry process), Exchange Requirements (which define the information to be exchanged), Exchange Requirements Models (which organize the information into Exchange Concepts that will linked to Concepts in the MVD and enable verification that all requirements have been satisfied), and a Generic BIM Guide (which documents guidance to the end user about what objects and data must be included in the BIM to be exchanged). Product specific versions of the BIM Guide will be developed later by vendors of certified software products.

Phase 2: MVD – A Model View Definitions document a subset of the IFC Model Specification that is required for the information exchanges defined in one or more related IDMs. As such, it is the ‘design’ for support of those information exchanges in software products. MVDs include three primary deliverables, each using standard formats. These are: MVD Overview/Description (which describes the scope of the MVD; especially the IDM (industry processes and information exchanges) that are addressed), MVD Diagrams (which define the MVD Concepts that will be used in the exchange, as well as the structure and relationships between these Concepts), and Concept Implementation Guidance (which defines the IFC entities used to exchange each concept and the Implementer agreements that general reduce the implementation scope that would otherwise be required by the extremely general IFC schema).

Phase 3: Software Implementation/Certification – Once MVDs are documented and posted, support for vendor implementation in their software products is required. This involves technical support for vendors who are implementing support for the IFC data exchange in their products as well as detailed, Concept by Concept testing to confirm whether their implementation is conforms to the MVD specifications. Results of Certification Testing are made public, so that end users can review them and the certification process is open. In most industries, software testing to assess conformance to a standard is outsourced. This is also been the case for both buildingSMART and members of BLIS – who have contracted testing by GTDS and Digital Alchemy, respectively.

Phase 4: BIM Validation – Certified software products do not ensure successful IFC based data exchange I projects. The end user must successful apply guidance documented in the BIM Guide for the exchange.

---

\(^1\) History about the transition of IAI to buildingSMART can be found on the bSI website: [http://www.buildingsmart.com](http://www.buildingsmart.com)

\(^2\) A good source for understanding the IFC model is the IFC wiki: [http://www.ifcwiki.org/index.php](http://www.ifcwiki.org/index.php)

\(^3\) An example IFC model sever project in BLIS was SABLE: [http://www.blis-project/~sable](http://www.blis-project/~sable)


\(^5\) [http://www.ifd-library.org](http://www.ifd-library.org)
Background and Acknowledgements

buildingSMART (then International Alliance for Interoperability) began with a process oriented approach, development activity for several years focused on developing technical data specifications for the Industry Foundation Classes (IFC). Initial implementations and compatibility between those implementations was inconsistent, resulting in end user disappointment.

In 1999, The BLIS Consortium was formed to address this reliability problem. In 2000, BLIS introduced the notion of Model Views -- as well-defined subsets of the IFC model that must be supported by software for specific end user scenarios. BLIS Model Views were defined in terms of reusable Exchange Concepts. Each Concept was defined and a binding to specific entities in the IFC model schema that are used to exchange that concept was documented. This purpose of this Exchange Concept documentation was to unambiguously define the implementation requirements such that implementations by different vendors would be consistent. The value of such documentation was proven in 2002-2003 as members of BLIS developed and certified some 35 applications for consistent support of 4 out of the 6 Model Views defined by BLIS.

At about the same time, buildingSMART formed the Implementers Support Group (ISG). ISG also focused on improving compatibility between applications by defining the Coordination View of IFC. As with BLIS, ISG certified several applications for support of the Coordination View over the past decade.

After the early work done by ISG and BLIS, two initiatives began in 2005. One group began the development of IDM, the other group (which grew out of BLIS) begin the development of a more formalized approach to model view definitions. The group developing the IDM focused on documenting industry processes and information exchange requirements. The group developing MVD focused on documenting information exchanges in a manner that could be directly implemented in software and that encouraged reuse of ‘information packages’ called Concepts. These Concepts eventually became the basis for Certification Testing and BIM Validation.

A large number of people and organizations have contributed to the content of this document, which can best be characterized as an attempt to harmonize and connect different ideas and efforts related to implementing support for the IFC based data exchange in software products, as well as the practical use of such software in building industry projects. Anyone active in this area has contributed in one way or another, especially people involved in buildingSMART and BLIS.

IDM acknowledgements: Jeffrey Wix played a central role in developing the methodology behind Information Delivery Manuals. Several other persons have contributed to the development: Janne Marit Aas-Jakobsen, Kjetil Espedokken, Ole Kristian Kvarsvik, Mark Bew, Lars Christensen, Dianne Davis, Bill East, Eilif Hjelseth, Ole Kristian Kvarsvik, Gang Lee, Thomas Liebich, Celson Lima, Henk Schaap, Richard See, Bjorn Stangeland, Rasso Steinmann and Justin Wong. buildingSMART activities in Norway have funded significant parts of this development.

MVD acknowledgements: Development of the official IFC Model View Definition format was initiated by a proposal from BLIS at ITM Summit #29 in Madrid, February 2005. That proposal and work completed between 2005 and 2008 was led by Jiri Hietanen (author of v1 & v2), and made possible through funding from the Finnish VBE2 project. In the first stage Kari Karstila and Jeff Wix contributed to the harmonization work between the approaches of BLIS, ProIT and IDM. Others who had significant influence include: Janne Marit Aas-Jakobsen, Kjetil Espedokken, Ole Kristian Kvarsvik and Sakari. Valuable feedback was provided by: Vladimir Bazjanac, Chuck Eastman, Kent Reed and Richard See from the BUILDINGSMART Technical Advisory Group. In addition a large number of people actively participated in related meetings, online training sessions and presentations, creating and refining the material which defines MVD. From 2008 to 2011, continuous improvement of MVD tools, templates and coordinating MVD Concepts database was funded and managed by The BLIS Consortium. Published MVD documentation has been continuously hosted on the BLIS website.

BLIS website. Managed by The BLIS Consortium. Published documentation has been continuously hosted on the continuous improvement of MVD tools, templates and coordinating MVD Concepts database was funded and managed by The BLIS Consortium. Published MVD documentation has been continuously hosted on the BLIS website.

6 Information about The BLIS Consortium can be found at: http://www.blis-project.org
7 Information about ISG can be found at: http://buildingsmart-tech.org/implementation
An Integrated Process

This work integrates IDM s and MVD, and identifies certification testing and end user BIM Validation based on requirements defined in IDMs and MVDs as necessary activities to insure industry use.

The IDM process is now an ISO Standard designed for use by industry workgroups. The industry workgroups define scope, workflows and requirements for Information Exchanges. Additionally, workgroup instructions have been produced in 2010-2011. This documentation includes templates for IDM narratives, workflow diagrams, and exchange requirements. The US documentation identifies reference standards, which add more structure to the narrative and scope documentation for IDMs. Additionally, tools for IDM automation are being developed.

There is a harmonization around the terms Exchange Concepts, Functional Parts, and Exchange Objects. In the future Exchange Concepts will be the preferred term. The definition:

Process Overview

The integrated process defined in the remainder of this document is largely based on agreements forged in meetings between August and October 2007. The process was first presented to the buildingSMART International Technical Management (ITM) committee during the November 2007 meetings in Brisbane, Australia.

An integrated IDM/MVD process has four phases and involves several participants.

![Figure 1 – Four Phase Integrated Process](image)

The following sub-sections will provide an overview of the participants, templates, tools and deliverables for each of these phases. More detail is provided in the sections later in the document – one each for each of these phases.

Requirements Definition – IDM – An industry led activity

The IDM process begins when AEC industry domain experts form a working group to develop an IDM for a specific process that would benefit from an IFC based information exchange. There are four deliverables in an IDM. The first task is to develop a consensus document or business use case for the target industry process. This use case identifies the process participants, points at which information content and format must be exchanged, and for what purpose. Next Process Maps are created using standard Business Process Modeling Notation (BPMN) templates. This notation also supports the nesting of sub-processes.

---

8 ISO 29481-1:2010 specifies a methodology and format for the development of an information delivery manual (IDM).
9 IDM Workgroup Instructions BSa 2011
10 Information about BPMN can be found at: [http://www.bpmn.org/](http://www.bpmn.org/) Are we going to use the document from Jeff Wix on this- with updates?
Exchange Requirements document the data to be included in each information exchange identified in the process map. These are normally documented in tabular or spreadsheet applications.

Finally, a series of entity relationship diagrams – call an Exchange Requirements Model (ERM) are developed for each high level object in the information exchange (e.g. Project, Site, Building, Building Story, Space, Wall, Door, Window, …). The building blocks of ERM diagrams are Exchange Concepts, each of which defines the information to be exchanged, the data type, minimums, maximums, defaults, etc. Existing Exchange Concepts are tracked in a coordinating database so that they can be reused across many IDMs and linked to MVD concepts that define a solution for exchanging the data in software. Through this link, we are able to verify if an IDM fully satisfies the information exchange requirements of the ERM.

Solution Design – MVD

A Model View Definition (MVD) document the subset of the IFC Model Specification that is required for the information exchanges defined in one or more related IDMs. The MVD is the ‘for software providers to support industry defined IDMs in software products.

MVDs include three primary deliverables, each using standard formats. These are: MVD Overview/Description (which describes the scope of the MVD) MVD Diagrams (which define the MVD Concepts that will be used in the exchange, as well as the structure and relationships between these Concepts), and Concept Implementation Guidance specifications (which define the IFC entities used to exchange each concept and the Implementer agreements that generally reduce the implementation scope that would otherwise be required by the IFC schema).

To date, MVDs have been defined primarily for implementation of information exchanges using IFC. The process and formats defined in this document covers the same scope. However, it is important to understand that MVDs can be used to define information exchange using any information model schema. This means that adopting organizations are free to experiment and prototype with other models, in addition to IFC.

In defining Model Views the goal is to find a useful balance between the wishes of the end user and the possibilities of software developers, and documenting the outcome clearly. The IFC Model View Definition Format is used for documenting this outcome.

The format must be well defined and unambiguous, but the format is only one part of what is needed. All of the following have been considered and addressed.

- **Format**: The type of data that needs to be captured and how that data is structured
- **Content**: The data that is needed in a specific case. For example the IFC Schema is content that is captured using the EXPRESS format and an IFC Model View Definition is content that is captured using the IFC Model View Definition format.
- **Process**: The roles and responsibilities of different involved parties, for example how a model view definition is endorsed, published, and how certification is organized.
- **Tools**: The tools used for creating content, e.g. defining concepts and concept diagrams, and managing the process of creating content. Tools are highly important, but the format itself must be independent from any specific tools.

Although the format is, in theory, independent from the other parts it must in practice support all of them. It is also clear that the format is not the full answer, but having a commonly agreed format is the starting point. Without a common format it is very difficult to reuse content and tools, or to define a clear process. You will learn about the templates for deliverables in the detailed section below.
Software Implementation and Certification

Even after requirements have been defined in the form of an IDM, and an implementable solution has been defined in the form of an MVD, the solution cannot be used in projects until it is supported by at least two software applications – the sender and receiver of the exchange. Support for vendors looking to implement support in their software products is important. Both ISG and BLIS have done a reasonable job of this in the past, but it can be improved, especially in the form of better documentation of implementation requirements, and in the form of tools that can be used by vendors during their development (see detailed section).

However, implementation support is not sufficient to ensure to end users that they will have a reliable exchange of BIM information in their projects. To ensure such reliable data exchange, third party testing and certification of software products is required.

Software certification testing for export requires that the vendor create certain test cases (BIMs) which include all data configurations required in the exchange. The Certification Test software then load the test case files and checks each object instance against the requirements defined in the ID/MVD. Conformance reports are generated by the testing software, reporting the success/failure of the exporting application in a manner that can be published for end users to review. Software certification testing for import is more complex because it requires evaluation of how the importing application has interpreted and used data in the exchange. At this time, import testing cannot be fully automated in the ways that export testing can, so the judgement of a testing expert is required.

To date, Certification Testing for IDMs/MVDs has been based on a combination of the following: MVD Diagrams XML, Concept level Implementation Guidance documents, and Business Rules from the ERM. This has involved human translation of these requirements into testing software. Earlier this year, a group in the German Speaking chapter of buildingSMART proposed an approach that would capture all of this information into a single XML file such that human interpretation of the requirements would be significantly reduced. The schema for this XML data is called MVD-XML. This should reduce any differences in testing that might have resulted from alternative implementations based on the current approach, but most likely will not be implemented in testing software until 2012.

This type of testing is very detailed and specialized. In virtually all other industries, such testing is done under contract or by appointment of companies/organizations that specialize in this type of testing. Both buildingSMART and BLIS are well advised to contract or appoint have historically contracted third party testing services that they believe will provide the necessary level of quality and reliability. Examples of third party certification testing that have resulted in certified applications used in building industry projects. buildingSMART has endorsed the Institut for Applied Building Informatics (iabi) at the University of Applied Sciences Munich campus to provide third party certification testing for the Coordination View of IFC (both versions 1 & 2). Similarly, several national property management organizations, including the General Services Administration (GSA) in the USA, Statsbygg in Norway, and Senate Properties in Finland have endorsed Digital Alchemy (a BLIS member company) to provide third party certification testing for 6 MVDs. These include: the Concept Design BIM 2010, the Design to Spatial Program Validation view, the Design to Energy Analysis View, the Design to Quantity Take-Off for Cost Estimating view, the Design to Circulation/Security Analysis view, and the Nordic Owners Energy Analysis View (see the BLIS IDM/MVD website\(^\text{11}\) for details on each of these).

BIM Validation and Use in Projects

Certification Testing is necessary for reliable data exchange, but not completely sufficient. This is because the software used to create the BIM to be exchanged is not the only determinant of the information in that exchange. The end user is just as important (if not more) in forming the data to be exchanged. If that end user does not follow instruction provided in the BIM Guide for the exchange (see overview of IDM above), then the exchange may not meet all requirements defined in the IDM. Therefore, even when certified applications are used, testing should be done for each individual BIM exchange in projects. This is called BIM Validation. This testing validates that the exporting application has met all software requirements, that the end user has used that software correctly, and that the user has met all end user requirements, including business rules, in creating the BIM to be exchanged.

As with Certification Testing for IDMs/MVDs, current testing is based on a combination of: MVD Diagrams XML, Concept level Implementation Guidance documents, and Business Rules from the ERM. The phase 3 version of MVD-XML, once it is complete, should improve consistency across multiple implementations of BIM Validation testing.

BIM Validation in projects will become increasingly important as project agreements begin to require the use of BIM exchange between project participants. In these cases, both sender and receiver will want to have third party validation that the contracted data exchange has been delivered.

\(^{11}\) Details for 28 MVDs can be reviewed at: [http://www.blis-project.org/IDM-MVD/](http://www.blis-project.org/IDM-MVD/)
As with software certification testing, BIM Validation is highly specialized. Both buildingSMART and BLIS are well advised to contract or endorse third party providers of such services.

Coordination of IDM/MVD Projects

As shown in figure 2, there are multiple stakeholders involved in a project from IDM definition through to BIM Validation. If projects are not coordinated, it is very possible that multiple solutions are developed for the same industry processes. Additionally, without coordination it is inevitable that the processes/exchanges that are defined will overlap, when they should be linking and complimentary. Both buildingSMART and BLIS have made attempts at coordination in the development of IDMs and MVDs. This paper defines a process through which various organizations (beginning with bSI and BLIS) can coordinate, but are still encouraged to innovate. That is: it is not necessary that all IDMs and all MVDs be controlled or managed by one organization. Instead, an eco-system in which many organizations are actively contributing to a growing collection of IDMs/MVDs should be encouraged and supported.

IDM/MVD Coordination in buildingSMART

To date, there has been activity at two levels of buildingSMART. buildingSMART International has long sponsored the development of the Coordination (Model) View, and has endorsed the iabi to perform certification testing. Additionally, the buildingSMART alliance (in North America) has been active in the development of IDMs/MVDs by member organizations. Notable examples include the US GSA, the Precast Concrete Institute, and the US Army Corps of Engineers. These IFC based information exchanges will be published for use in industry through the US National BIM Standard (NBIMS) after suitable early implementations have been tested and verified.

So buildingSMART is active in promoting IDM/MVD standards at both the international and national (regional) levels. Member organizations at both these levels are encouraged to ensure coordination of their IDM/MVD projects with the appropriate level of buildingSMART. This is done not to control these projects, but to enable coordination, consistency, and maximum benefit to the building industry.

Contact points for such coordination include:

- **IDMs** -- International IDM Coordinator or National/Region IDM Coordinator
- **MVDs** -- International MVD Coordinator or National/Region MVD Coordinator
- **Software Implementation** -- ISG Leader or National/Region Implementation Coordinator

Contact information for each of these roles can be found at the various buildingSMART websites.

IDM/MVD Coordination in BLIS

IDM/MVD Coordination in The BLIS Consortium is done mainly through the BLIS IDM/MVD website, and through a series of tools that have been developed to make such coordination a natural part of the process.

Contact points for such coordination include:

- **IDMs** -- IDM Coordinator
- **MVDs** -- MVD Coordinator
- **Software Implementation** -- Project Certification Testing organization or MVD Coordinator

Contact information for each of these roles can be found at the BLIS IDM/MVD website.

IDM/MVD Coordination in Other Organizations

Any industry consortium or association can facilitate the development if IDMs and MVDs in order to enable IFC based BIM data exchange for it member companies. Examples that have already been active in this arena include:

- US GSA, Statsbygg (Norway), Senate Properties (Finland)
- American Institute of Steel Construction (AISC)
- Pankow Foundation
- Precast Concrete Institute (PCI)

Coordination of IDM/MVD development in these projects has been done through both buildingSMART and BLIS.

12 The BLIS Consortium coordinating IDM/MVD website is: [http://www.blis-project.org/IDM-MVD/](http://www.blis-project.org/IDM-MVD/)
Organizational Endorsement

Endorsement of IDM/MVDs
It should be clear that IDM and MVD can be developed for many different levels of standardization across the global AECOO industries, including:

- International – including buildingSMART, BLIS, or other international industry associations
- National – including buildingSMART or other national industry associations
- Regional – including regional industry associations or groups
- Organizational – including large companies or groups of companies doing business together

Use of the process defined in this document should be encouraged at all of these levels. Those that are most used will naturally tend to float to the top – through industry use and growing software support.

Additionally, organizations at any of these levels may choose to publish their IDM/MVDs on their organizational websites, or may choose to use site already developed by buildingSMART or BLIS. Any of these options is fine. The collective goal should be to increase the use of IFC based BIM data exchange in building industry projects.

Endorsement of Certification and BIM Validation Testing

As stated elsewhere, it is highly unlikely (and possibly ill-advised) that any of these organizations try to develop Certification and BIM Validation testing themselves. It is much more practical for them to provide unambiguous definitions for what must be tested to companies that specialize in such certification testing. The recently proposed MVD-XML is an example of how such testing requirements could be provided to such testing agencies.

Both buildingSMART and BLIS should seek out such testing agencies and work to develop an eco-system of end users, software vendors, and member organizations that will encourage the testing agency to develop the certification testing speculatively. Funding of such development is cost prohibitive for both buildingSMART and BLIS. The two examples cited above (GTDS for Coordination View and Digital Alchemy Testing Service for several other IDM/MVDs) prove that this is possible.

This approach is, of course, also available to organizations at the other levels of standardization listed above. In fact, endorsement at these various levels is a path to developing the eco-system that will be necessary to support such speculative development.
Developing an IDM

Experience Required for IDM Development
The development of the IDM can start after a relevant use case has been identified and the possibility for adoption of the digital support workflow among the relevant parties has been investigated and commitment from software companies to make the potential necessary development has been guaranteed.

As the main purpose of an IDM is to identify information requirement in a use case it is important to involved people with domain specific knowledge. Since an existing workflow without support from digital data exchange can be improved by making adjustments to the workflow it is necessary to include people in the development of the IDM there are aware of the benefits of a digital supported workflow.

It the IDM should become an official buildingSMART IDM it is necessary to include several organization potential from different countries depending on the level of acceptance. If the IDM should become an multinational recognized IDM the documentation have to be accessible in English.

Although the work related to IDM is based on user needs it can anyhow be useful to have experts related to the development that are able to judge if the required information are specified in the preferred schema or not. In the case of the buildingSMART data model IFC it should be recognized that if the needed information not already are included in the specification it can be a time consuming process before the required information are a part of an official IFC release.

It is not from a technical or use case point of view necessary to have business managers or change agents involved in the team, but it is important to remember to have somebody to be responsible bringing the idea of a digital supported workflow into practice since implementation in most circumstances not are straight forward process.

IDM Coordination
During the development of an IDM it is important:

- to follow the IDM methodology
- communicate and coordinate the work with other existing or ongoing IDM development teams
- insure thoroughly review is preformed
- to incorporate the approval process for official recognition of the IDM if the developed IDM should become an official IDM
- be aware if the IDM should be supported by the use of IFC whether extensions are needed or not.

Forming a Domain Group
- Is this different from "Experience Required for IDM Development" ?

IDM Deliverables
The main deliveries from an IDM development process are process map, exchange requirements, exchange requirement model and BIM Guide. Business Rules and Validation?
Figure 2: Principal Components of IDM, where the Exchange Requirement Model is a configuration of the Exchange Concepts that satisfy all Exchange Requirements.

Process Map
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.

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 3: Reference processes in the IDM technical architecture.

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.

The preferred approach to developing a process map within IDM is to use the Business Process Modelling Notation (BPMN). (REDUCED VERSION?)

(Add information on process maps, like in the MVD section)

**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.

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

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 5 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 6 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.

(Add information on process maps, like in the MVD section)

**Exchange Requirements 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.

An exchange concept 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 an exchange concept.
Each exchange concept 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 exchange concept may also relate to many exchange requirements.

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

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 exchange concepts that can then be compiled to provide the exchange requirement model schema.

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

(Add information on process maps, like in the MVD section)

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 concepts 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 concept, this is coded in the logical form appropriate to the manner in which the attribute/property is expressed.

In IDM, a technical expression of an exchange requirement can be derived by compiling the constituent concepts. 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 it 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 ex-
change 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

(Add information on process maps, like in the MVD section)

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 vali-
dation tests are carried out. This is because different sets of business rules may be applied to the same ex-
change 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 veri-
fied
- making comparisons between software applications fulfilling the same objectives
  (when compared using the same tests)
- estimating reliability

(Add information on process maps, like in the MVD section)

Generic BIM Guide (incl. business rules)

From Abu Dhabi slides:

General guidance to users on how an exchange requirement should be met using Building Information Mod-
elling.
This should be independent of application (may form the basis of local standards)

Application Specific
Specific guidance to users – how to meet the exchange requirement using a specific product

---------- content moved from MVD for ERM begins here -----------

Reusable Exchange Concepts

The main enabling mechanism for re-use of ‘packages of data to be exchanged’ is Exchange Concepts. All
advanced view definition formats make use of this idea; including ProIT, pre-integrated IDM and pre-
integrated MVD. The core idea is: commonly useful packages of information are identified in the Exchange
Requirements Model as Exchange Concepts. The technical solution for exchanging them in software
(bound to IFC model entities) is called MVD Concepts.

Exchange Concepts are independent from any IFC Model View Definition. Technically an IFC Model View
Definition is created by choosing (or defining) a group of concepts and defining their relationships. For ex-
ample a “rectangular profile” concept could be selected into an IFC Model View Definition, but defined to
only be used with beams and columns, not spaces and walls. Another form of re-use is to separate the idea
of a concept from the IFC binding of that concept. This makes it possible to re-use the same concept ideas
when the underlying IFC Model Specification changes. For example the idea of a “space name” does not
An Integrated Process for Delivering IFC Based Data Exchange

change if moved at some point from IfcSpace.LongName to some other location in the IFC Model Specification.

In our integrated IDM-MVD process Exchange Concepts (independent of IFC) are color-coded with blue and MVD Concepts (bound to an IFC schema) with orange.

Although not a part of the IDM/MVD formats, software tools are highly important for re-using definitions. These formats defines a system which makes it possible to re-use definitions, but tools can make it much easier to know what has already been defined. Tools also help share the definitions with large groups making it less likely that the same definitions are reinvented. Example Tool Sets are listed in Appendix C

IDM Coordination

The final stage in developing an IDM is development of the Exchange Requirements Model (ERM). The ERM is a diagrammatic representation of the Exchange Requirements – which show both the packages of data to be exchanged (Exchange Concepts) and the relationships between them – from an end user perspective.

It is important to study existing concepts and to re-use them whenever possible. Also the structure of existing definition (patterns) should be re-used as much as possible.

When the ERM is done, the IDM is complete and development of an MVD that can be implemented in software can be considered.

ERM Description

The purpose of the ERM description is to document the scope of the Exchange. It defines an overview of the exchange and is used as a discussion paper in the process of defining the ERM. The description should be limited to one page.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Title field&gt;</td>
<td>The name of the MVD</td>
</tr>
<tr>
<td>Reference</td>
<td>The reference number of the MVD.</td>
</tr>
<tr>
<td></td>
<td>&lt;Author ID&gt;-&lt;MVD number&gt;</td>
</tr>
<tr>
<td>Version</td>
<td>The sequential version number of the MVD</td>
</tr>
<tr>
<td>Status</td>
<td>The status of the MVD.</td>
</tr>
<tr>
<td></td>
<td>Sample, Draft, Proposal, Candidate, Official or</td>
</tr>
<tr>
<td></td>
<td>Deprecated</td>
</tr>
<tr>
<td>History</td>
<td>The history of the MVD, e.g. a version history</td>
</tr>
<tr>
<td>Document Owner</td>
<td>The document does not contain a field for copyright. The document owner is the person or organization responsible for maintaining the document, i.e. the only one allowed to make changes to the document. Should contain some contact information, e.g. email address.</td>
</tr>
</tbody>
</table>

Figure 7 MS Word template for the ERM Description

13 ERM_Description.dot
ERM Diagrams

ERM/MVD Diagrams Relationship

As part of the integrating IDM and MVD, the IFC independent concept diagrams were moved to IDM and became the Exchange Requirements Model (ERM). The ERM serves as a machine readable version of requirements defined in the IDM -- that could be used later for BIM Validation.

Exchange Concepts in the ERM define re-usable packages of information to be exchanged. An ERM diagram defines the scope of data to be exchanged about a top level 'Variable Concept'.

MVD Concepts define a way to realize the Exchange Concept in software data exchange. An MVD diagram also defines the scope of data to be exchanged about the top level 'Variable Concept'. The top level concepts generally correspond between ERM and MVD diagrams, but occasionally some additional top-level concepts may be introduced in the MVD for software implementation purposes.

Both ERM and MVD will have ‘Overview’ sheets that describe the entire scope of the ERM/MVD. In this section we are focused on the ERM.

ERM diagrams define which of the re-usable Exchange Concepts are used in a specific ERM, and the relationships between those concepts. A diagram may for example define that walls can have a classification reference. ERM diagram headers are always shaded blue so that they are readily recognized. ERM diagrams should be understandable to industry practitioners (i.e. people without knowledge of IFCs or software implementation).

Exchange Concepts define data sets to be exchanged in generic terms. They may even be used to define concepts that are not exchanged through IFCs.
ERM Diagram Specifics

The format for ERM diagrams and configuration is defined by an XML schema\textsuperscript{14}. These diagrams can be created manually, or using tool sets provided by vendors (see appendix C).

The XML schema for ERM diagrams supports three different styles, which may be combined into the same XML dataset

- **Definition**: the concepts used in a diagram and their relationships in the context of that diagram.
- **Configuration**: The status of the concepts (ON/OFF) and diagram specific comments for concepts.
- **Layout**: The position, visibility and other layout related settings of concepts in a diagram. The layout is typically specific to the coordinate system of the diagramming application (e.g. the MS Visio). Layouts are not part of the ERM format per se, but are important to the presentation of ERM diagrams.

This division makes it possible to create several configurations and layouts for the same definition.

Since the format for ERM diagrams is defined in an XML schema, there is no official page size or orientation. Large diagrams will require one ‘page’. That is: there is no provision for splitting a diagram over several pages. Some vendor diagramming tools have features to help accommodate large diagrams by hiding parts of the diagram in certain contexts. Such settings are saved in the diagram layout, as explained above.

A separate ERM Diagram is created for each Variable (or high level) Concept in the ERM.

---

**Figure 9** Template for generic view diagram

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagram name</td>
<td>The name of the diagram is the name of the variable concept of the diagram. The name is shown in the title.</td>
</tr>
<tr>
<td>View Name</td>
<td>The name of the MVD</td>
</tr>
<tr>
<td>Application name</td>
<td>The name of the software application for which the diagram is made.</td>
</tr>
<tr>
<td>Application version</td>
<td>The version of the software application for which the diagram is made.</td>
</tr>
<tr>
<td>Exchange type</td>
<td>Generic, Import, Export or Roundtrip</td>
</tr>
</tbody>
</table>

\textsuperscript{14} ViewDefinition.xsd
A diagram defines which concepts are used in a view and the relationships between those concepts. Static, group and adapter concepts may be placed on the right side of the variable concept. Connectors in the diagram always point from left to right. Circular connections between concepts are not allowed and each concept may only be connected to one 'parent concept'.

A concept may be marked as optional if it is not essential to the exchange, but desirable. Concepts are marked as optional by turning them off (light grey, dashed line boxes). Software may still be certified for support of the view if they don’t support optional concepts, but may be more preferred by end users if they do support them.

Diagrams may be configured using two mechanisms: making concepts optional (turning them off) and adding comments to the concepts. Making concepts optional reduces scope. Commenting is used to make diagrams more specific or to enhance understanding about relationships or data included in the exchange. In addition diagrams may contain any text or graphical elements, but such elements will not be captured in the XML format generated from the diagram.

Large diagrams may be placed on one page by hiding concepts. Hiding a concept does not mean that it is turned off. Hiding is used purely for layout purposes.

ERM Concept Definition

Exchange Concepts
Exchange Concepts enable clear definition and reuse of data packages, as well as unambiguous requirements specification for that will be addressed by the corresponding MVD concepts(s) to be supported in software applications (see MVD section).
Variable Concept

The same variable concepts can be used in different MVDs, but their content may vary. Hence the variable concept must be configured separately for each MVD. This configuration is done by creating a diagram in which other concepts (group and static) are attached to the variable concept.

Examples: space in quantity take-off, wall in HVAC design

Group Concept

Group concepts provide structure for the concept diagrams by grouping together static concepts and/or other group concepts.

Examples: space properties, wall geometry

Static Concept

Static concepts remain the same in all scenarios in which they are used. They can be re-used without modification because they don’t contain any options.

Examples: space number, bounding box geometry

Each concept has an ID, which uniquely identifies the concept. The name is not used as the ID because concepts may be translated into different languages. The ID has the following format.

<Author ID>-<Concept Number>

For example; TEMP-001, ABC-123

Please note: The Author ID “MVC” (Model View Concept) is assigned only to concepts that have been proven and are shared across multiple IDMs, as determined by the IDM Coordinator.

In MVDXML, each concept is given a ‘fully qualified name’, which identifies it in the context of the diagram. This name is created by iterating from the concept through all parent concepts to the variable concept and finally to the MVD. The fully qualified name is used when definitions and configurations are compared with each other.

If the example above was from MVD with the Reference “TEST-01”, the fully qualified name for “Space Number” would be.

Test-01:TEMP-001:TEMP-002:TEMP-003

Concept Documentation

There is a separate description document for each IFC independent and IFC release specific concept. The official format for the documents is PDF. A Microsoft Word template is provided for creating the documents but any other software or system may be used as well.

In documents based on the template any field marked with <… field> should be edited through the document properties of the MS Word document.

Exchange Concept Description

The IFC independent concept description contains the detailed definition of the concept.
### IFC Independent Concept Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Title field&gt;</td>
<td>The name of the concept</td>
</tr>
<tr>
<td>Reference</td>
<td>The reference number of the concept. &lt;Author ID&gt;-&lt;Concept number&gt;</td>
</tr>
<tr>
<td>Version</td>
<td>The sequential version number of the concept</td>
</tr>
<tr>
<td>Status</td>
<td>The status of the concept. Sample, Draft, Proposal, Candidate, Official or Deprecated</td>
</tr>
<tr>
<td>Relationships</td>
<td>Relationships to other concepts</td>
</tr>
<tr>
<td></td>
<td>• Extends : the group concept the concept is based on</td>
</tr>
<tr>
<td>History</td>
<td>The history of the concept, e.g. a version history</td>
</tr>
<tr>
<td>Document Owner</td>
<td>The document does not contain a field for copyright. The document owner is the person or organization responsible for maintaining the document, i.e. the only one allowed to make changes to the document. Should contain some contact information, e.g. email address.</td>
</tr>
<tr>
<td>Description</td>
<td>The free form description of the concept, preferably only one page long. If a copyright is asserted this can be done in the description field.</td>
</tr>
</tbody>
</table>

**Transition to MVD**

<to be developed>

---

15 ConceptDescription_IFCIndependent.dot
Developing an MVD

MVD Background
Traditionally IFC Model View Definitions have been understood as subsets of the IFC Model Specification and have been defined primarily for IFC implementation purposes. The format defined in this document covers the same scope. However, it is important to understand the connections it has to a larger picture, in which requirements come from the value chain of the end user and the primary role of IFC Model View Definitions is to ensure that IFC implementations support those requirements.

For definition of IFC Model Views the goal must be to define an exchange of IFC data that will meet the end user’s needs, as defined in the IDM, and is implementable for both sending and receiving software applications. The Model View Definition Format is designed to document that implementable exchange with the least ambiguity possible.

The format must be well defined and unambiguous, but the format is only one part of what is needed.

- **Format**: The type of data that needs to be captured and how that data is structured
- **Content**: The data that is needed in a specific case. For example the IFC Schema is content that is captured using the EXPRESS format and an IFC Model View Definition is content that is captured using the IFC Model View Definition format.
- **Process**: The roles and responsibilities of different involved parties, for example how a model view definition becomes official and how certification is organized.
- **Tools**: The tools used for creating content, e.g. defining concepts and concept diagrams, and managing the process of creating content. Tools are highly important, but the format itself must be independent from any specific tools.

Although the format is, in theory, independent from the other parts it must in practice support all of them. It is also clear that the format is not the full answer, but having a commonly agreed format is the starting point. Without a common format it is very difficult to reuse content and tools, or to define a clear process.

This format is based on the IFC Model View Definition and related formats developed by the BLIS\(^{16}\), ProIT\(^{17}\) and IDM\(^{18}\) projects and it has been developed and validated by the people behind these efforts. It was originally targeted only at defining the scope and details of IFC implementations, and for providing a way to certify such implementations. In essence MVD provides the specification for IFC based data exchange implementation, and certification tests how well implementations comply with this specification.

However, in addition to certification the question of BIM Validation has become increasingly important. BIM Validation does not test the software application, but rather, how the implementation is applied by a software user in a project. Certification may for example test that a classification reference can be exchanged between software products, whereas BIM Validation may test that a classification reference is provided by a software user, and that the provided reference is from an agreed local classification system.

In 2008, two major improvements were added to the IDM and MVD definition formats, tools, and process. The first was the inclusion of an IFC based technical solution for BIM Validation. This is accomplished by attaching business rules to the concepts defining the required data exchange capabilities. Certification is done against concepts, validation against business rules. The concept in the example above could be ‘Classification reference’, and the business rule ‘Must use classification system xyz’.

The second major improvement was a further simplification of the interface between IDM and MVD. This was done in two areas. First the language for defining end user requirements for data exchange between software products was harmonized. This enables a simpler and more effective way to communicate such requirements to software implementers, and on the other side, to communicate the capabilities of IFC implementations to software users. Secondly moving the IFC based technical solution for validation from IDM to MVD made IDM a pure requirement definition methodology and MVD a pure methodology for IFC based technical solutions. While this change does not provide any additional features to the combination of these two methodologies, it does help significantly in explaining and applying the integrated IDM/MVD methodology.

\(^{16}\) http://www.blis-project.org
\(^{17}\) http://virtual.vtt.fi/proit
\(^{18}\) http://idm.buildingsmart.com
The context for MVD

Figure 11 Overall architecture

The goal of any IFC related activity must be deployment in projects. In order to reach deployment the requirements of deployment must be known and there must be a technical solution for satisfying these requirements. The architecture shown here has a loose coupling between the different parts. This means, for example, that there can be many technical solutions for satisfying the same data exchange requirement, or many different requirement documentations based on the same deployment needs. As a theoretical development process the requirements of deployment are captured, a technical solution is developed to satisfy these requirements and finally the technical solution is deployed. In reality the process is often different, mainly because the development of technical solutions is commercial activity and guided by many other factors in addition to well defined and document end user requirements.

Figure 12 The role of MVD (with IDM as an example for the requirement and deployment definition methodology)

The role of MVD is to provide an IFC based technical solution for end user requirements captured and documented using any requirements definition methodology. However, MVD has a well defined interface to IDM, which provides several important benefits.

1. The requirements defined using the IDM methodology, i.e. Exchange Requirements (ER) share the same IFC independent requirement definition format with MVD. This makes it easy to merge several ERs (which define data exchange between actors in a project) into one MVD (which defines data exchange between software application types). This connection helps ensure that the requirements for software implementation (MVD) cover the requirements of data exchange between actors in a project (ER).

2. MVD is used in software certification as the requirements specification for IFC based data exchange. The results of certification are documented using the IFC independent requirement definition format, which is shared with IDM ERs. This makes it easy to relate the certification results of a specific software application to the requirements for data exchange between actors in a project, i.e. knowing if that software application can be used successfully in a given end user scenario. When data exchange in a project is defined in a contract, it is of utmost importance to know, if a selected software application can be used for satisfying the terms of the contract.
3. The requirements for data validation are captured in IDM ERs as business rules (BR). The business rules can be mapped to the IFC based technical solution for data validation and implemented in data validation software. This helps ensure that the technical solution for data validation matches the requirements for data validation.

![Diagram](image)

**Figure 13** Steps needed for reaching deployment\(^{19}\) *(modified)*

The figure above shows the different steps that are needed for creating IFC based interoperable solutions that are successfully deployed in AEC/FM projects. It is like a 'task list' for all the things that must be taken care of. The picture is shaped like a pyramid, because the shortcomings of any level limit the possibilities of the levels above it. For example shortcomings in IFC implementations would naturally limit their deployment in projects. Also the possibilities build into the lower levels are not automatically available on the higher levels. For example expanding the scope of the IFC specification does not automatically mean that the new scope is available in IFC implementations. In short, it is necessary to build a solid foundation for the deployment of IFC based solutions. The IFC specification, i.e. the IFC schema and its documentation, is naturally at the core of any IFC based solution. In addition to a solid foundation the structure must reach deployment before being valuable to the industry.

**Goals for MVD**

The main goal of MVD is to enable high quality IFC implementations that satisfy a given set of data exchange requirements defined in one or more IDMs. The MVD format should further satisfy the following requirements.

- Enable data exchanges, as defined in the IDM process and formats, in building industry projects. The MVD process does not change these requirements, but may refine and merge data exchange requirements into packages that are meaningful from the viewpoint of software implementation.

- Provide a way for software developers to implement meaningful IFC support in software without wasting resources. Implementing an MVD should be the easiest way to implement IFC support in software.

- In order for IFC to become a mainstream data exchange solution, implementing IFC support must not require face-to-face meetings or attendance in workshops. This applies only to implementing support for an already agreed data exchange scenario. Face-to-face meetings can still be used in the process of defining data exchange scenarios.

- Certification must provide useful information about the capabilities and limitations of IFC based data exchange. It is important that industry practitioners understand the results of certification testing (i.e. what can and cannot be exchanged between the sending and receiving applications).

**Reusable MVD Concepts**

As discussed in the IDM section, reusable definitions of commonly useful packages of information are identified in the Exchange Requirements Model as Exchange Concepts. The technical solution for exchanging them in software (bound to IFC model entities) is MVD Concepts.

In our integrated IDM-MVD process Exchange Concepts (independent of IFC) are color-coded with blue and MVD Concepts (bound to an IFC schema) with orange.

Also mentioned above, software tools for developing IDMs and MVDs are essential to making reuse of concepts easy. Example Tool Sets are listed in Appendix C

---

\(^{19}\) Based on “The Interoperability Pyramid” (Hietanen, 2003)
Experience Required for MVD Development
Development of an MVD is a rather specialized process. It requires in-depth knowledge of the information model(s) for which bindings will be defined, as well as a good understanding of the requirements (and industry process) described in the IDM.

MVD Coordination
In order for IFC BIM exchange to be widely used in the building industry, many teams should be developing IDMs and MVDs – addressing requirements in all regions of the world. Therefore it is important that our process encourage and support distributed development. On the other hand, such distributed development can easily lead to reinventing concepts instead of reusing them. This means that coordination across many teams developing IDMs and MVDs is very important. It must be easy for any developer to find existing concepts and for them to register new concepts for others to find and reuse.

MVDs are designed to make it possible for end users to exchange in real world business processes. In the transition from Exchange Requirements and associated ERM to MVD it is necessary to translate from end user domain definitions to software implementation definitions. The first deliverable in making this transition is a one page description of the new MVD. This one page description should be reviewed by all teams involved in developing MVDs – so that, if such an MVD already exists or is close, the teams involved can consider: is it better to expand the existing MVD or do develop a new one.

Since MVDs are of no use unless they are implemented in software, the developing team should also ensure that software vendors are willing (if not eager) to implement the MVD. In fact software implementer involvement is essential to the MVD development process. One of the key aspects of MVD Concept definition is the Implementation Agreements section that limits implementation requirements where the IFC model schema defines them vague or broad terms. For example: the IFC schema may allow for 5 different geometry representations for a given concept, yet the implementers may agree that limiting this to 2 alternative representations is sufficient for the Concept in the context of this MVD.

When MVD diagrams have been developed for all variable concepts included in the MVD, and implementation guidance documents have been developed for all implementable concepts included in those diagrams, the MVD is complete. Development of Certification Testing can then be addressed (see next major section).

MVD Deliverables
There are several documents that define an MVD. These include:

- IFC Release Specific MVD Description
- IFC Release Binding MVD Diagrams
- MVD Concepts – Implementation Guidance documents
- MVDXML – a machine readable definition of the subset of IFC used in the MVD

Each of these is described and examples given in the following sections.
**IFC Release Specific MVD Description**

The purpose of the IFC release specific MVD description is to document any general decisions that were made regarding the IFC binding.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;IFC Release&gt;</code></td>
<td>The IFC release for which the binding is defined</td>
</tr>
<tr>
<td><code>&lt;Title &gt;</code></td>
<td>The name of the MVD</td>
</tr>
<tr>
<td>Reference</td>
<td>The reference number of the MVD</td>
</tr>
<tr>
<td></td>
<td><code>&lt;Author ID&gt;-&lt;MVD Number&gt;</code></td>
</tr>
<tr>
<td>Version</td>
<td>The sequential version number of the MVD</td>
</tr>
<tr>
<td>Status</td>
<td>The status of the MVD. Sample, Draft, Proposal, Candidate, Official or Deprecated</td>
</tr>
<tr>
<td>History</td>
<td>Any history specific to the IFC binding</td>
</tr>
<tr>
<td>Document Owner</td>
<td>The document does not contain a field for copyright. The document owner is the person or organization responsible for maintaining the document, i.e. the only one allowed to make changes to the document. Should contain some contact information, e.g. email address.</td>
</tr>
<tr>
<td>Description</td>
<td>1. Which version of the generic view definition is being used</td>
</tr>
<tr>
<td></td>
<td>2. Basic principles applied when mapping the generic view to the specific IFC release, including implementer’s agreements.</td>
</tr>
<tr>
<td></td>
<td>3. Limitations relative to the generic definition</td>
</tr>
<tr>
<td>NOTE:</td>
<td>If a copyright is asserted this can be done in the description field.</td>
</tr>
</tbody>
</table>

---

20 MVDDescription_IFCReleaseSpecific.dot
MVD Diagrams

ERM/MVD Diagrams Relationship

As described above, the IFC independent concept diagrams in previous versions of MVD were moved to IDM to become the Exchange Requirements Model (ERM). The ERM serves as a machine readable version of requirements defined in the IDM -- that could be used later for BIM Validation.

Exchange Concepts in the ERM define re-usable packages of information to be exchanged. An ERM diagram defines the scope of data to be exchanged about a top level 'Variable Concept'.

MVD Concepts define a way to realize the Exchange Concept in software data exchange. An MVD diagram also defines the scope of data to be exchanged about the top level 'Variable Concept'. The top level concepts generally correspond between ERM and MVD diagrams, but occasionally some additional top-level concepts may be introduced in the MVD for software implementation purposes.

Both ERM and MVD will have 'Overview' sheets that describe the entire scope of the ERM/MVD. In this section we are focused on the MVD.

---

For each Exchange Concept in the ERM, there will be one or more MVD concepts which define a way to exchange the data defined in that concept in software. In fact, the MVD concept has a 'Implements' relationship to the Exchange Concept it implements. MVD diagrams are targeted at software developers, especially people writing software code.

MVD Concepts define the binding of their corresponding Exchange Concepts into a specific IFC release. They define how the IFC Model Specification is used for exchanging the required data, e.g. that a classification reference is exchanged using the IfcClassificationNotation object. Each supported IFC release will have its own binding documentation, because the details of how the same data is captured may change between IFC releases.

MVD Diagram Specifics

The format for MVD diagrams and configuration is defined by an XML schema\(^\text{21}\). These diagrams can be created manually, or using tool sets provided by vendors (see appendix C).

The XML schema for MVD diagrams supports three different styles, which may be combined into the same XML dataset

- **Definition**: the concepts used in a diagram and their relationships in the context of that diagram.

---

\(^{21}\) ViewDefinition.xsd
• **Configuration:** The status of the concepts (ON/OFF) and diagram specific comments for concepts.

• **Layout:** The position, visibility and other layout related settings of concepts in a diagram. The layout is typically specific to the coordinate system of the diagramming application (e.g. the MS Visio). Layouts are not part of the MVD format per se, but are important to the presentation of MVD diagrams.

This division makes it possible to create several configurations and layouts for the same definition.

Since the format for MVD diagrams is defined in an XML schema, there is no official page size or orientation. Large diagrams will require one ‘page’. That is: there is no provision for splitting a diagram over several pages. Some vendor diagramming tools have features to help accommodate large diagrams by hiding parts of the diagram in certain contexts. Such settings are saved in the diagram layout, as explained above.

A separate MVD Diagram is created for each Variable (or high level) Concept in the MVD.

---

**Figure 16 Template for IFC release specific view diagram**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagram name</td>
<td>The name of the diagram is the name of the variable concept of the diagram. The name is shown in the title.</td>
</tr>
<tr>
<td>IFC Release</td>
<td>The IFC release the diagram is defining the binding for. The IFC release is shown in the title.</td>
</tr>
<tr>
<td>View Name</td>
<td>The name of the IFC Model View Definition</td>
</tr>
<tr>
<td>Application name</td>
<td>The name of the software application for which the diagram is made.</td>
</tr>
<tr>
<td>(optional)</td>
<td></td>
</tr>
<tr>
<td>Application version</td>
<td>The version of the software application for which the diagram is made.</td>
</tr>
<tr>
<td>(optional)</td>
<td></td>
</tr>
<tr>
<td>Exchange type</td>
<td>Generic, Import, Export or Roundtrip</td>
</tr>
<tr>
<td>Diagram status</td>
<td>Sample, Import, Proposal, Candidate, Official or Deprecated</td>
</tr>
<tr>
<td>Diagram version</td>
<td>The sequential version number of the diagram</td>
</tr>
<tr>
<td>Diagram date</td>
<td>The data the version of the diagram was completed</td>
</tr>
<tr>
<td>Diagram authors</td>
<td>The authors of the diagram</td>
</tr>
<tr>
<td>Document Owner</td>
<td>The person or organization responsible for maintaining the dia-</td>
</tr>
</tbody>
</table>
A diagram defines which concepts are used in a view and the relationships between those concepts. Static, group and adapter concepts may be placed on the right side of the variable concept. Connectors in the diagram always point from left to right. Circular connections between concepts are not allowed and each concept may only be connected to one ‘parent concept’.

A concept may be marked as optional if it is not essential to the exchange, but desirable. Concepts are marked as optional by turning them off (light grey, dashed line boxes). Software may still be certified for support of the view if they don’t support optional concepts, but may be more preferred by end users if they do support them.

Diagrams may be configured using two mechanisms: making concepts optional (turning them off) and adding comments to the concepts. Making concepts optional reduces scope. Commenting is used to make diagrams more specific or to enhance understanding about relationships or data included in the exchange. In addition diagrams may contain any text or graphical elements, but such elements will not be captured in the XML format generated from the diagram.

Large diagrams may be placed on one page by hiding concepts. Hiding a concept does not mean that it is turned off. Hiding is used purely for layout purposes.
MVD Concept Definition

MVD Concepts
MVD Concepts address the data exchange requirements defined in the corresponding Exchange Concept (see ERM subsection of IDM).

<table>
<thead>
<tr>
<th>IFC Release Specific Concepts</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable Concept</strong></td>
<td>The IFC binding of a variable concept implements a generic variable concept with the same name. Example: the IFC Binding of the variable concept “Wall” is “Wall”, not “Wall standard case”</td>
</tr>
<tr>
<td><strong>Adapter Concept</strong></td>
<td>Adapter concepts are reusable parts of the IFC model specification that connect static concepts to a variable concept. There is no correspondence between adapter concept and IFC release independent group concept. Instead, adapter concepts provide a proposal how to structure software code in IFC implementations for reaching maximal code reuse. Examples: classification assignment, property set assignment</td>
</tr>
<tr>
<td><strong>Static Concept</strong></td>
<td>The IFC binding of a static concept implements one or more generic static concepts. The names of generic and IFC binding static concepts don’t have to match. The documentation of the IFC binding contains a detailed definition how to apply the IFC Model Specification. Example: “GUID” implements the IFC release independent concept “Software Internal ID”.</td>
</tr>
</tbody>
</table>

Each concept has an ID, which uniquely identifies the concept. The name is not used as the ID because concepts may be translated into different languages. The ID has the following format.

<Author ID>-<Concept Number>

For example; TEMP-001, ABC-123

Please note: The Author ID “MVC” (Model View Concept) is assigned only to concepts that have been proven and are shared across multiple MVDs, as determined by the MVD Coordinator.

In MVDXML, each concept is given a ‘fully qualified name’, which identifies it in the context of the diagram. This name is created by iterating from the concept through all parent concepts to the variable concept and finally to the MVD. The fully qualified name is used when definitions and configurations are compared with each other.

If the example above was from MVD with the Reference “TEST-01”, the fully qualified name for “Space Number” would be.

Test-01:TEMP-001:TEMP-002:TEMP-003
Concept Documentation
There is a separate description document for each IFC independent and IFC release specific concept.

The official format for the documents is PDF. A Microsoft Word template is provided for creating the documents but any other software or system may be used as well.

In documents based on the template any field marked with <… field> should be edited through the document properties of the MS Word document.

MVD Concept Description
The IFC release specific concept description contains the binding of the concept to specific IFC release.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;IFC Release field&gt;</td>
<td>The IFC release for which the binding is defined</td>
</tr>
<tr>
<td>&lt;Title field&gt;</td>
<td>The name of the concept</td>
</tr>
<tr>
<td>Reference</td>
<td>The reference number of the concept</td>
</tr>
<tr>
<td>Version</td>
<td>The sequential version number of the concept</td>
</tr>
<tr>
<td>Status</td>
<td>The status of the concept. Sample, Draft, Proposal, Candidate, Official or Deprecated</td>
</tr>
<tr>
<td>Relationships</td>
<td>Relationships to other concepts</td>
</tr>
<tr>
<td></td>
<td>• Implements : the IFC release independent concept implemented by the IFC release specific concept</td>
</tr>
<tr>
<td></td>
<td>• Extends : the adapter concept the concept is based on</td>
</tr>
<tr>
<td>History</td>
<td>The history of the concept, e.g. a version history</td>
</tr>
<tr>
<td>Document Owner</td>
<td>The document does not contain a field for copyright. The document owner is the person or organization responsible for maintaining the document, i.e. the only one allowed to make changes to the document. Should contain some contact information, e.g. email address.</td>
</tr>
</tbody>
</table>

Figure 17 MS Word template for the IFC release specific concept description document

22 ConceptDescription_IFCReleaseSpecific.dot
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage in MVD Diagram</td>
<td>The place of the concept in an MVD diagram, for example.</td>
</tr>
<tr>
<td>Instantiation Diagram</td>
<td>The IFC entities, which need to be instantiated for the concept and the relationships between the entities. Instantiation diagrams may be freely commented and may contain clarifying drawings. Instantiation diagrams from other concepts may be inherited and further specified. Any IFC entity that requires implementation agreements, see definition below, are marked with light yellow.</td>
</tr>
</tbody>
</table>

**Classification Association**

- IfcRelAssociatesClassification
  - GlobalId
  - OwnerHistory
  - Name
  - Description
  - RelatedObjects
  - RelatingClassification

**Classification Notation**

- IfcClassificationNotation
  - NotationFacets
    - IfcClassificationNotationFacet
      - NotationValue

There is no official format for instantiation diagrams, but the use of the notation in the example above is encouraged. MS Visio templates are provided for instance diagrams.

**Implementation Agreements**

Often the IFC specification contains ambiguity about how it should be applied in specific cases, i.e. it defines more than one way of doing the same thing. In such cases implementation agreements must be used for defining the one agreed way to do a single thing.

The basic format for implementation agreements is a table containing the attributes of an IFC entity used in the IFC binding of the concept (see instantiation diagram above). Example:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Implementation agreements</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalId</td>
<td>Must be given, but may change</td>
</tr>
<tr>
<td>OwnerHistory</td>
<td>Must be given, but may contain dummy data</td>
</tr>
<tr>
<td>Name</td>
<td>Reserved</td>
</tr>
<tr>
<td>Description</td>
<td>Reserved</td>
</tr>
<tr>
<td>RelatingSpace</td>
<td>No agreements needed</td>
</tr>
<tr>
<td>RelatedBuildingElement</td>
<td>No agreements at this point</td>
</tr>
<tr>
<td>ConnectionGeometry</td>
<td>If provided, must be IfcConnectionSurfaceGeometry</td>
</tr>
<tr>
<td>PhysicalOrVirtualBoundary</td>
<td>&lt;Open&gt;</td>
</tr>
<tr>
<td>InternalOrExternalBoundary</td>
<td>&lt;Open&gt;</td>
</tr>
</tbody>
</table>

The following standard agreements can be used:

- **<Open>** The authors of the concepts have not dealt with the attribute yet. Should appear only in incomplete descriptions
- **No agreements needed** The IFC specification is unambiguous and no agreements are needed
- **No agreements at this point** There are no agreements defined in this concept, but agreements are made in concepts that inherit from the concept.
- **Reserved** There are no agreements currently, but the attribute is reserved for future use. This is usually used with labels and other descriptive strings.
The concept may contain any number of additional information. Such information is not part of the official definition, but can make it easier to understand and implement the concept or help in creating certification test cases etc. Examples:

- EXPRESS-G diagram
- EXPRESS sub schema
- UML diagram
- Sample files

**Concept Based Implementation Guidance Specifications**

<To be developed by RichSee>

**Implementation agreements**

In the IFC Model View Definition format there is no separate documentation for implementation agreements. All implementation agreements are captured in the IFC binding of the MVDs.

The high level description of an IFC binding contains all agreements which are not specific to the use of individual concepts. This would cover cases like: use IFC2x2 property sets in IFC2x implementations. There will probably not be many agreements on this level, maybe even none.

A static concept has to be fully supported and there are no options inside a static concept. For software users the capabilities of IFC implementations are easier to understand if static concepts have a large scope. For implementations large concepts can be problematic because even software created for the same purpose is very different and a large concept may be discriminating.

![Figure 18 Large concepts vs. small concepts](image)

In the example above the software user would like to know if steel profiles can be exchanged. However, if an application can support all other steel profiles but not 'Z Shape', the certification results would say that the application doesn't support the exchange of steel profiles. Whenever this is a problem concepts have to be defined on a more granular level.

Detailed agreements about how the IFC Model Specification is used are captured in the IFC bindings of concepts. This would cover cases like: the name of a space is exchange using IfcSpace.LongName, the number of a space using IfcSpace.Name and all spaces must be contained by a building storey.

Implementation agreements provide the specification that must be followed when implementing IFC support in software. Certification checks that this specification has been followed. Certification test cases cannot be generated automatically from the IFC Model View Definition format, but the format allows capturing all information necessary for creating test cases manually.

**MVD Schema**

<To be developed by RichSee>

- MVD-XML
- Tools
Lifecycle of an MVD

MVDs have a life cycle, which spans from the idea (originating e.g. from IDM Exchange Requirements) to the time the definition is superseded by another definition.

Idea: Someone has an idea for a new MVD, documents and publishes the idea for others to review.

Draft: Some interested party creates and documents a new MVD or extends an existing MVD. Any person or organization is allowed to do this without restrictions or limitation.

Proposal: If the author of an MVD wants to seek endorsement from an organization or standards group like buildingSMART or BLIS. When such a proposal is accepted by the endorsing organization, the MVD status will be set to Proposal – and the context of that proposal (i.e. proposal to buildingSMART or BLIS) will be made clear.

Candidate: When an MVD has been submitted to an organization for endorsement. That organization should review it using the following criteria.

• Has it been documented using the standard MVD format?
• Does it make correct use of the IFC Model Specification?
• Is there overlap or conflicts with existing MVDs?

The proposal may be refined based on the feedback from the endorsing organization. Once the MVD satisfies the set criteria it has Candidate status – for the endorsing organization only. As with Proposal status, the context of Candidate Status will be clearly shown on the MVD web site (i.e. Candidate as buildingSMART or BLIS endorsed MVD).

Official: The distinction between Candidate and Official MVD status is that Certification testing is available for the latter. Software certification must be organized by the organization endorsing the MVD, and must be openly available to any software vendor (at cost to the vendor). In most cases, Certification testing will be done under a contract organized by the endorsing organization (e.g. GTDS by buildingSMART international and Digital Alchemy by GSA). When more than one software application has been certified, the MVD status will be elevated to Official. After this any software can apply for certification against exactly the same definition.

Deprecated: When an MVD is superseded by another definition it should be Deprecated. In practice this means that certification testing is no longer available for the deprecated MVD.

Transition to Implementation/Testing

<To be developed by RichSee>
Software Implementation and Certification

Motivating Software Vendors to support the Exchanges

Implementation Support
<To be developed by RichSee>

Certification Testing
<To be developed by RichSee>
To date, there are two such organizations performing certification testing under contract: Institute for applied Building Informatics (Germany)(for buildingSMART International – Coordination View 2) and Digital Alchemy (USA)(for the US General Services Administration(USA), Statsbygg (Norway), and Senate Properties (Finland).

<more coming from RichSee>

Encouraging Industry to Develop IDMs/MVDs
<To be developed by RichSee>
BIM Validation - Using Certified Apps in Projects

MVD for data validation

The original purpose of MVD was to provide a specification for the IFC based technical solution for data exchange between software applications. This includes the scope and details of IFC implementations, and enabling certification based on this specification. The goal was to create IFC based, reliable and useful data exchange capabilities for industry practitioners for either creating or consuming BIM data.

However, exactly the same format can also be used for the IFC based technical solution for data validation. This system has been defined for IDM based exchange requirements (ER), which are documented as an exchange requirement model (ERM).

In the original approach the concepts from several ERMs are rolled together into one MVD, which corresponds to the exchange between software application types. When MVD is used for data validation the business rules (BR) of an ERM are used instead, and rolled together into a meaningful package from the viewpoint of data validation software. The BRs become data validation concepts, which are defined like any other concept. In the IFC binding of these data validation concepts use the constraint model of the IFC specification.

When used in projects, data validation software reads in two files (or data sets); the design data and the constraint data. Both data are based on an MVD, the design data on a ‘design MVD’ and the constraint data on a ‘validation MVD’. The implementations of both can be certified using the IFC certification process.

<more coming from RichSee>
Appendix A: IDM Templates
- Reuse information from [1,2]

Appendix B: MVD Templates
- ????

Appendix C: IDM/MVD Development Tool Kits
- BLIS Toolkit for IDM/MVD – See: http://www.blis-project.org/IDM-MVD

Appendix D: Testing Services
- MVD Certification Testing Services
  - GTDS – (put link here)
  - Digital Alchemy – http://DigitalAlchemyPro.com
- BIM Validation Services for End Users
  - Digital Alchemy – http://DigitalAlchemyPro.com

Appendix D: Example Projects
- Design to Spatial Program Analysis
- Design to Energy Analysis
- Precast Concrete Industry Exchanges
Parking Lot

All of the following has value, but has not been placed as of yet …

Methodology from requirement to implemented solution

As specified above the integrated methodology is based on a sequence starting from identification of user requirements to a technical specification, implementation of a software solution or configuration continuing to software certification and data validation.

![Figure 2: Transformation of needs into operational solutions](image)

In the following figure shows a sample of a flow of IDM and MVD to software certification and data validation.

![Figure 19: Flow of information from User Requirements in IDM to MVD that are implemented in software tools. Data Validation is done in accordance to the IDM.](image)

Since some user requirements may be included in existing Model View Definitions an Information Delivery Manual does not automatically lead to new software implementations. Data Validation should always be based on Information Delivery Manual that may restrict the values of the properties and which objects there are mandatory, option or don't may occur in an exchange scenario.
Figure 20: User Requirements which already are covered by an existing MVD and certification. Data Validation should anyhow be compared to the original requirement in the IDM.

Figure 21: MVD encapsulating several exchange requirements - DO WE AGREE? I got a valid comment from Norway that it should be possible to certify a product like dRofus without dRofus has to implement a "full/big/composed Model View Definition

Information Delivery Manual
Exchange requirements between identified actors are captured for a specific process in the lifecycle of a building or facility. The flow of information are captured in process diagrams and data will be modelled in an Exchange Requirement Model. The Exchange Model are based on exchange concepts which are schema independent.

Input:
- Information needs expressed by domain experts.

Output:
- Process diagrams following the BPMN graphical representation
- Business Rules as Express/text (?)
- Exchange Requirement Model defined by exchange concepts (format, idmXML?)
Model View Definitions
A Model View Definition transforms an Exchange Requirement Model into a schema specific specification e.g. IFC4. There will exist one schema specific MVD per schema e.g.

Input
- Exchange Requirement Model from an Information Delivery Manual

Output
- Schema specific representation of exchange concepts in mvdXML-format
- What about business rules?
- What about Validation?
- What about Simple ifcXML?

Software Implementation
The step is related to the software development that is need in order to support the exchange of the exchange concepts specified in a Model View Definition.

Input
- Schema specific representation of exchange concepts in mvdXML-format
- What about business rules?
- What about Validation?
- What about Simple ifcXML?

Output
- Software that is able to support data exchange of exchange concepts according to the schemas specified in the mvdXML-format

Software Certification
Specification of the content that is needed to be supported by the certified software. The general requirements to how the certification should be performed are documented in separate procedures.

Certified according to IDM or MVD?

Input
- Schema specific representation of exchange concepts in mvdXML-format
- What about business rules?
- What about Validation?

Output
- Approval or rejection of software certification certificate

Data Validation
Validation of the content in data files are according to the requirement in IDM and MVD. The IDM specifies which data there are mandatory or optional, and the business rules specifies additional rules that are controlling the values of the properties or the logic between the elements.

Input
- Data file that should be validated
- Schema specific representation of exchange concepts in mvdXML-format
- What about business rules?
- What about Validation?

Output
- Approval or rejection of whether the data file is according to the specification or not.
IDM Vision and Goals
It is 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.

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

The goals of the IDM are to:

- define the processes within the AEC/FM project lifecycle for which users require information exchange
- 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

There are some general trends related to the goal of reaching deployment of IFC based solutions. These trends deal with creating and using new possibilities, the required technical skills and number of people involved, and questions of international vs. local.

The development work on layers below the deployment layer is aimed at enabling and improving deployment. In this sense all efforts should ultimately be driven by the requirements of deployment; without deployment the whole system has no reason to exist. However, inventing new possibilities requires innovation and developing new possibilities requires taking risks. It would be incorrect to assume that development is driven primarily by requirements from people working on the deployment level. Typically people working on the lower levels must have a vision, take a risk and develop new possibilities, with the hope that these will be accepted and adopted by people on the deployment level.

The number of people involved will increase dramatically when IFC based interoperability becomes standard practice. Because of this it is important to build a system, in which the required level of technical skills decreases proportionally whenever the number of people involved increases. This may be as dramatic as having a dozen people working on the IFC specification and several million working in deployment, i.e. using IFC based solutions in their daily work. The success of deployment cannot be allowed to be directly dependent on a small group of IFC experts.

IFC is an international standard and construction activity is local, which means that a transition from international to local must happen between these two. Software is increasingly international and IFC model view definitions, which sit between the IFC specification and IFC implementations, are by this logic also international. Of course international software is localized to specific markets and there is still a large number of
purely local software, which means there can also be demand for local model view definitions. Interoperability know-how has many international elements, but applying interoperable solutions in a local context requires information about local practices and legislation. Finally deployment is by nature local, because buildings are constructed and maintained locally.

Central control can be strong in an international effort that involves only a small number of people, such as the development of the IFC specification. In the other extreme nobody would allow an international organization, such as the buildingSMART, to control construction projects. In the case of MVD the balance for buildingSMART is to open up the definition of MVD content, but to coordinate this content to ensure consistent use of the IFC specification.

Definitions and configurations
The IFC Model View Definition format makes extensive use of definitions and configurations. Definitions capture a range of possibilities and configurations capture how those possibilities are used in a specific case.

![Figure 23 Definitions and configurations](image)

A configuration is always based on a single definition. However, there can be several configurations of the same definition. For example software certification is done against a single MVD (definition) and the result of the certification is a configuration of that MVD for each certified software product.

A configuration is created by first choosing which possibilities captured in the definition are in scope for the configuration. This typically leads to a reduction of the original scope. A configuration may never increase the scope of the definition. For the remaining scope a configuration adds rules and agreements about how that scope is to be used. Such rules and agreements must not be in conflict with the original definitions. For example each MVD is a configuration of the IFC Model Specification. Each MVD defines a subset of the IFC Model Specification and adds new rules, called implementation agreements, to it. These implementation agreements must not be in conflict with the IFC schema or its documentation.

![Figure 24 Comparing configurations of the same definition](image)

It is possible to compare different configurations of the same definition. For example comparing the MVD configurations (which are produced by certification) of a read and write certified software, will reveal the type of data that can be successfully exchanged between these two software products. If the definitions and configurations are documented using a computer interpretable format (schema) it is possible to use software for performing such comparisons.
In the figure above all blue boxes are comparable, because they are all based on the same definition (MVD A). This common definition can be created by merging business process driven Exchange Requirements that can be satisfied using the same type of software. The result is a collection of concepts and relationships between the concepts. Following this system any blue box can be compared with any other blue box, which provides answers to some practical questions.

- What data can be exchanged between two software products using IFCs and what are the limitations of this data exchange? Compare the configurations of the software products for the same IFC Model View Definition.
- How well does a software product support an Exchange Requirement? Compare the configuration of the software product with the configuration of the Exchange Requirement Model.
- What new data is required when moving from one project stage to another? Compare the configurations of two Exchange Requirement Models.

**Roadmap for MVD definition work**

Ultimately all MVD definition work should be guided by the needs of deployment, i.e. by data that is needed in business processes. This is not a new idea to buildingSMART. From the beginning, it has relied on the work of domain teams to define the scope of the IFC Model Specifications. The work of these domain teams is best captured in volume 1 of the IFC R2.0 documentation. After 1999 there has been IFC implementation and software certification activity based on view definitions from different sources. Because MVD definition work is not starting from scratch there is need for a roadmap for the work.

1. Where are we today? Already implemented IFC views and related implementer’s agreements should be documented using the official MVD format.

2. How can we make the most out of the existing possibilities? This generation of MVDs should mainly assume existing software and the current IFC Model Specification. This combination can already provide much better value than is available through existing IFC implementations.

3. What are the ultimate possibilities of IFC based exchange? This is a forward looking generation of MVDs, which require changes to existing software and/or the current IFC Model Specification. This stage assumes that software vendors are putting major effort into improving their software from the interoperability viewpoint. Typically strong customer demand is a prerequisite for this.

- This roadmap does not mean a strict sequential approach, i.e. the different stages can be overlapping. The purpose is to give a general idea how to proceed towards the target process.

---

23 IFC R2.0 “Vol. 1 AEC/FM Processes Supported by IFC” (See, 1999)
2 New exchange requirements are created and documented when people/organizations have a common vision for improving or re-engineering old processes. The need to have formal contracts about data content drives the formal definition of exchange requirements.

3 If needed, customers demand software capable of delivering/consuming the data defined by the new exchange requirements.

4 If needed, existing IFC Model View Definitions are expanded and new definitions are created and documented to server the needs of the new/expanded exchange requirements.

5 If needed, the IFC Model Specification is expanded to enable the new/expanded IFC Model View Definitions.

--- END ----- Removed from MVD Intro sections – after Reusable Concepts ---------------