Industry Foundation Classes
Architecture and Development Guidelines

J.Wix[1] and T.Liebich[2]

Introduction

This paper discusses significant technical developments which have taken place within the International Alliance for interoperability (IAI) in the development of release 1.5 of the Industry Foundation Classes (IFC). In particular, three elements of the development are presented:-

- The IFC Model Architecture describes the framework developed to achieve interoperability between different disciplines (or domains) within the AEC/FM industry.
- IFC Object Model documentation describes progress in presenting information about IFC model developments in a form which can be easily understood by industry.
- Guidelines for the Development of IFCs describes the development of an easily understood procedure which can be used by the members of the IAI to assist them in the development of IFCs.

Background

The ideal of sharing information between computer applications in the AEC/FM industry has long been the target of research and development effort. It is only recently that the technology which allows us to contemplate achieving this ideal has become available in terms of both hardware and software. However, the greatest move forward has been in the development of tools and techniques which allow the inclusion of the actual users of the technology into the development process.

The open sharing of information without regard to the hardware or software applications in use has come to be known as interoperability. It places emphasis on the value of information and how it is used rather than on the systems which use it. In this sense, interoperability can be said to empower the owner and user of the information.

Interoperability requires that concepts which are common between different software applications are understood to be common and declared accordingly. This understanding needs to be present within the computer systems running interoperable software and not just by their human operators. Since computer systems do not have the power of interpretation expected of a human user, development of the necessary understanding means that:-

- Names given to classes correspond between applications.
- The meanings assigned to these names is consistent between software applications.
- The sharing of classes between applications is unambiguous.
- The relationships defined between the objects (including inheritance relationships) are compatible.

To fulfil the needs of industry, concepts of interoperability have to be developed collaboratively, by end users and software developers acting together and in an open environment. It is for these reasons that Industry Foundation Classes are being developed by the IAI.
The scope defined by the IAI for the IFC Object Model is "enabling interoperability between AEC/FM applications from different software vendors". The AEC/FM industry is, by its nature, fragmented and distributed. This industry also encompasses a very large

[1] Jeffrey Wix is the Senior Partner of Jeffrey Wix Consulting based in the United Kingdom.
[2] Dr. Thomas Liebich is a Partner of Junge and Liebich based in Germany.

model space. In understanding this, it becomes obvious that the IFC Object model must be decomposed into smaller and more manageable components. This provides the rationale for decomposition of information models presented in the IFC Object Model Architecture..

The IFC Object Model Architecture

The IFC Object Model Architecture has been developed using a set of principles governing it's organisation and structure. These principles focus on basic requirements and can be summarised as:

- Provide a modular structure to the model.
- Provide a framework for sharing information between different disciplines within the AEC/FM industry.
- Ease the continued maintenance and development of the model.
- Enable information modelers to reuse model components
- Enable software authors to reuse software components
- Facilitate the provision of upward compatibility between model versions

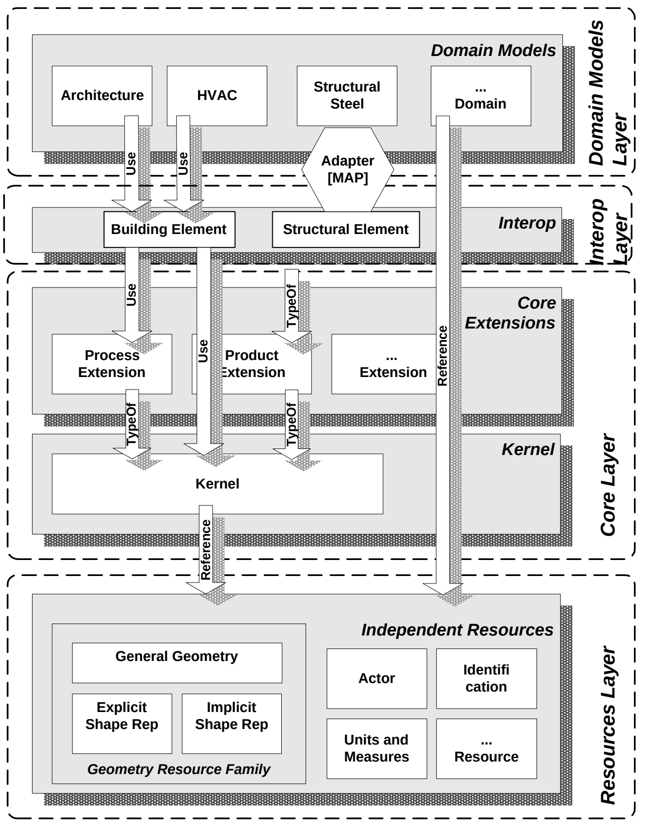


Figure 1: IFC Object Model Architecture

Layered Approach

The IFC Object Model architecture provides a modular structure for the development of model components. There are four conceptual layers within the architecture, which use a strict referencing hierarchy. Within each layer a number of grouped modules exists.

The Resource Layer

Resources are ideas which do not rely on classes within the Kernel for their existence. For instance, geometry can be created as a resource and then used to describe the shape of an object defined in the Kernel or Domain Extension models.

Resources represent individual business concepts. For instance, basic information concerning cost is within the cost resource. Classes within the Core or Domain Models which need cost reference this resource.

Resources form the lowest layer in IFC Model Architecture and can be used or referenced by all classes in the other layers.

The Core Layer

The Core layer consists of two parts namely the Kernel and the Core Extensions.

Kernel

The Kernel provides all the basic concepts shared between domains in AEC/FM projects. It also determines the model structure and decomposition. Concepts defined in the kernel are abstracted to a high level. Fundamental concepts concerning the provision of objects, relationships, type definitions, attributes and roles are also included.

The Kernel is the means by which a bridge is created between individual domain requirements and provides the platform for all model extensions. The constructs that form the Kernel are very generic and are *not* AEC/FM specific, although they will only be used for AEC/FM purposes due to the specialisation by Core Extensions. The Kernel constructs will be included as a mandatory part of all IFC implementations.

The Kernel is the foundation of the Core Model. Kernel classes may reference classes in the Independent Resources but may not reference those in the other parts of the Core model or any classes in the Domain Models. The use of Resources is facilitated by well defined interfaces within resource schemata. Thus, the design detail for any particular resource will be hidden from referencing classes.

Core Extensions

Core Extensions provide extensions to concepts rooted in the Kernel. They are the first refinement layer for abstract Kernel constructs. More specifically, they extend Kernel constructs for use within the AEC/FM industry. Each of these is a specialisation of object classes defined in the Kernel. Primary relationships and roles are also defined within the Core Extensions.

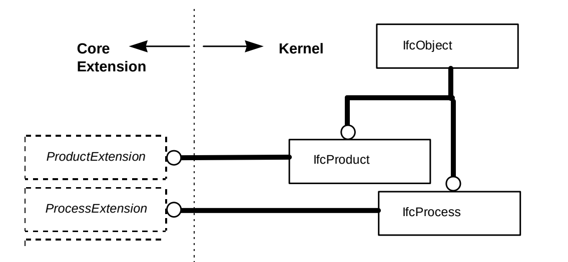


Figure 2: Core Extensions from Kernel Classes

A class defined within a Core Extension may be used or referenced by classes within the Interoperability layer but not by a class within the Kernel or Resources. References between Core Extensions have to be defined very carefully in a way that allows the selection of a singular Core Extension without destroying data integrity by invalid external references.

The Interoperability Layer

The Interoperability layer provides specialised, well defined interfaces for one or more Domain Models. This provides for outsourcing of Domain Model development whilst retaining control over the key structuring and framework requirements of interoperability..

Adapters

Whereas the dominant approach of IFCs is expected to be models developed from the IFC Core and Resources, a number of information models have been developed (and are likely to continue to develop) which are outside the IFC development framework. Many of these models fulfil domain needs successfully. An objective is to provide a means by which such models share in the overall interoperability of models within the AEC/FM industry. Thus, the Architecture incorporates the concept of Adapters which enable external models to gain access to IFC functionality via the Interoperability layer. Technically, this is achieved through the use of

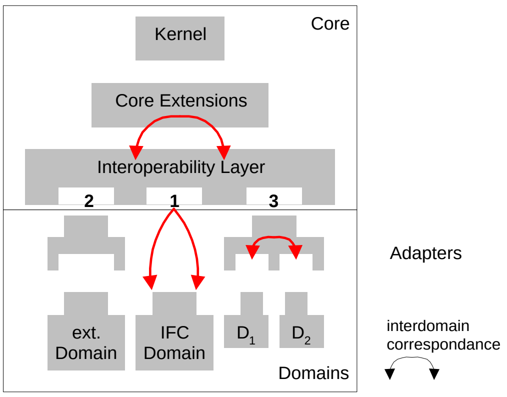


Figure 3: Interoperability Layer and Adapters

formal mapping languages (such as EXPRESS-X or the CSTB XP-Rule system)

In figure 3, points at the Interoperability layer fulfil the following functions:-
1. Inheritance by the Domain Specification from the Interoperability layer.
2. Interpretation of an external Domain Specification to classes in the Interoperability Layer.
3. Adapter providing sharing between external domain models which itself is interpreted to the Interoperability layer.

Domain Specifications

Domain Specifications provide further detail for an AEC domain process or a type of application. Each is a separate model which may use or reference any class defined in the Core and Resource layers. Examples are Architecture, HVAC, FM, Structural Engineering etc.

Documentation for the IFC Model

Documentation of IFC models uses HTML format where hot links are created between the various aspects of the model. The technique adopted uses a formal convention for naming and setting out of files and requires that parts of files contain embedded comments set up in a particular format.

The rationale behind the development of this form of documentation is that of ensuring that all parts are maintained consistently and that this consistency can be easily checked. It has come about from lessons learned during the IFC Release 1.0 development process where the collation of documentation took far longer than anyone had anticipated simply because of the volume of paper and information involved. Trying to keep several independent documents with independent authors in harmony with each other proved to be a ‘difficult’ task.

Document Parts in HTML

All parts of the object model are rendered into HTML format or a format which is compatible for viewing via a Web browser (such as PDF). Word documents contain comments which, when converted to HTML, can be interpreted by the PERL tool as requiring hot links to other documentation parts. EXPRESS files and the classes contained within them are always named according to their IFC name; this makes creation of hot links quite easy.

EXPRESS-G diagrams are converted into the PDF format which can then be viewed by the Web browser. PDF is a supported format so that again, hot linking to the graphics files is possible.

All classes within the EXPRESS listing incorporate access to their semantic definitions (which incorporates the semantic definitions of attributes) and graphics describing the usage scenario can also be incorporated into the EXPRESS listing. The facility for this is achieved by providing for special style of EXPRESS commenting. A similar technique is also used for provision of the MIDL interface code within EXPRESS; a facility which is otherwise not available.

The PERL Script

The key to the creation of the HTML documentation is a set of software functions written in the PERL language specifically for the purpose. Once all the documentation has been created and formed into the appropriate format, the PERL script can be executed. This ensures that all documentation is placed into the appropriate place and that all the relevant hot links and HTML document formatting are dealt with.

The IFC Model Development Guidelines

The IFC Model Development Guidelines provide guidance on the various stages of definition and development of Industry Foundation Classes. The emphasis is on providing guidance to **industry domain experts** who may not be familiar with formal software development methodologies.

Development Sequence

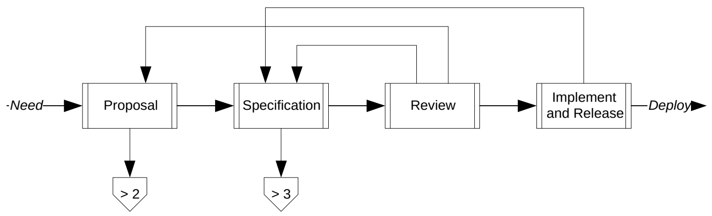


Figure 4: Overall Development Process

The development sequence of IFCs follows a sequence of progressive development stages. Of these stages, Proposal / Specification / Review are discussed in the Guidelines. Implement and Release is out of scope being subject to separate Implementation Guidelines.

Spiral Development

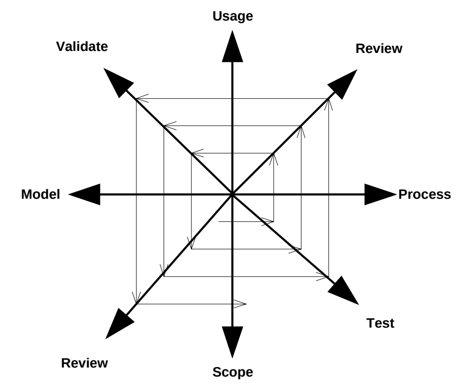


Figure 5: Spiral Development

Whilst the sequence identified appears to be linear, it does in fact comprise a cyclical development which incorporates concepts of review of fitness at various points and significant feedback both within a project stage and between project stages. The spiral development designated in the illustration below shows increasing detail as various cycles are undertaken. The diagram does not show specific feedback at the stages although it can be assumed to be present.

Overlapping Development

The different development steps will overlap in time, since a new IFC release will already be in the AEC process definition phase when the current release is in the pilot implementation phase.

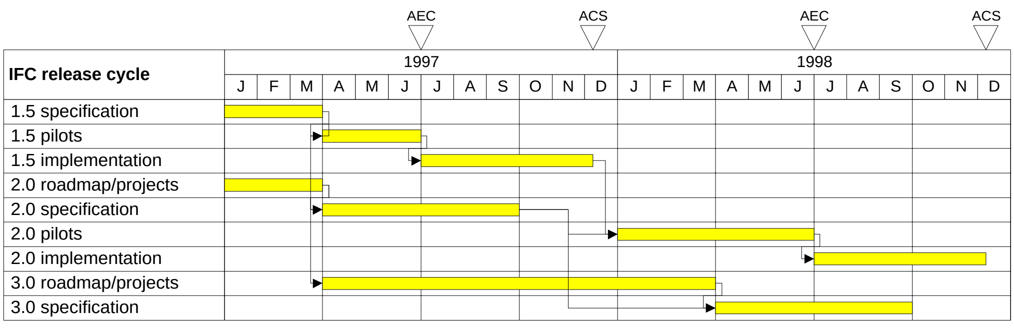


Figure 6: Time Overlapping of IFC Development Release Cycles

Proposal

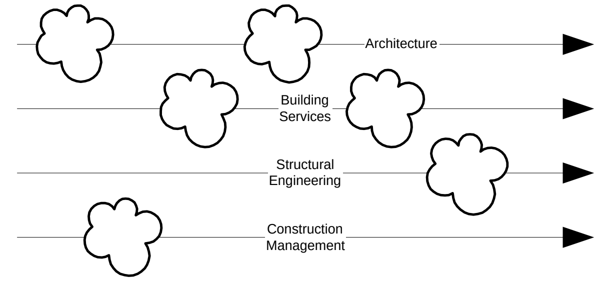


Figure 7: Non Touching and Not Very Interoperable Processes

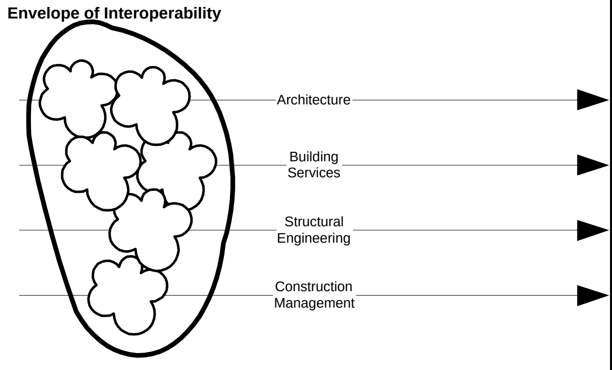


Figure 8: Touching Processes Within an 'Envelope of Interoperability'

Prior to commencing work on the development of an IFC specification, a proposal needs to be provided to the International Technical Management committee (ITM) of the intended development. The need for the proposal is to ensure that:-

- The intended development fulfills an identified industry need.

Whilst it is understood that an industry need may be identified within one Chapter, the objective is to seek out industry needs across all Chapters so that IFCs can support global concepts of interoperability wherever possible.

- The intended development, when considered in conjunction with other intended development from various domains and Chapters, provides a coherent extension to the current IFC capability.

It is not intended to limit the development of IFC specifications, the ITM wish to encourage the development of coherent sets of specifications which promote interoperability across domains. Therefore, the ITM actively looks for industry processes which can operate in conjunction with each other rather than processes which exist in isolation.

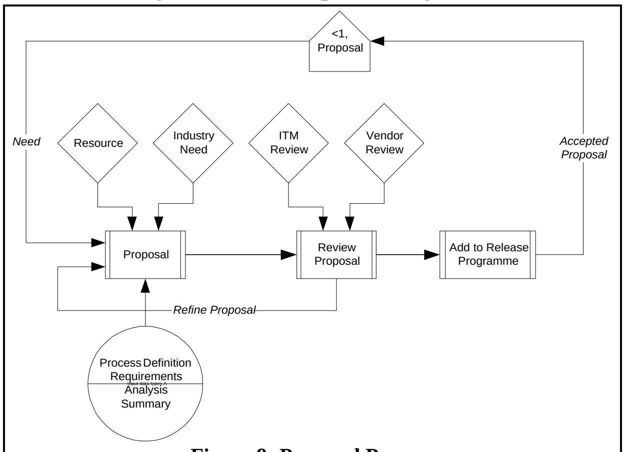


Figure 9: Proposal Process

- The resource required to ensure complete development of the specification supporting the defined industry process is available.

The ITM will, wherever possible, assist the development of a proposal by seeking additional resources to support development from Chapters other than that initiating the proposal.

Specification

All of the tasks to be undertaken during the specification stage of development build on work already completed during the proposal stage. Preliminary detail for usage scenario, process model, scope and an assessment of classes which will be incorporated into the IFC Object Model will have been completed. By using the tabular proposal layout, it will be possible to develop the specification to completion.

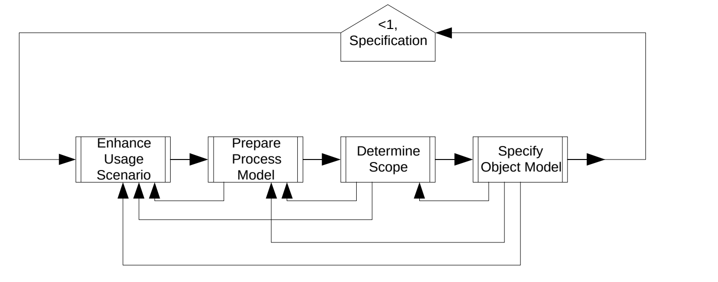


Figure 10: Specification Process

An important part of specification development which will not have been anticipated during the proposal stage is the synthesis of the object model for a given process with other specifications under development and the already existing IFC Object Model.

Enhance Usage Scenario

At this stage, the specification does not exist. The usage scenario is a description which provides a primary means for identifying its elements. The following rules should be observed in its creation:-

1 *Be assertive*. A usage scenario should provide a set of assertions which can be modelled and implemented e.g. A building is located on a site.
2 *Be clear* about the ideas being captured by the usage requirement. In the above, building and site are clearly identifiable as classes.
3 *Be specific* about relationships. In the above, the relationship ' is located on ' is captured.
4 *Be precise* concerning numerical constraints which exist. Frequently, this can be directly understood from the statement in the usage scenario. In the above, 'a building' identifies the fact that only one building is concerned whilst ' a site' recognises that only one site is concerned.

The usage scenario should also contain information enabling identification of the following items:-

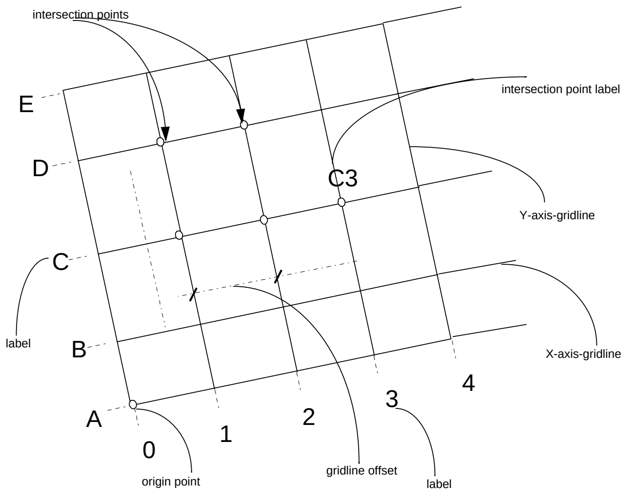


Figure 11: Example Usage Scenario Diagram for Grid

- Identify attributes including the units in which the attribute is measured e.g. A pump has a pump duty which is given as the mass flow rate (litre/sec) against the resistance of the pipework (N/m2).
- Identify required information which can be obtained from other domains, e.g. wall dimensions are normally obtained from the Architect and information which you expect to provide to other domains.
- The provision of drawings and sketches can add significantly to the quality and completeness of the usage scenario. They help to improve international understanding and can help the modeller and software implementers to understand how to realise the classes and relationships expressed.
- A vital aspect in developing IFC specifications is ensuring that the terms used to name classes and relationships are consistent within a specification and between specifications. Once the classes have been determined, their names should be semantically defined. Wherever possible, definitions should be derived from a validated industry glossary. That preferred is British Standard BS6100: Glossary of Building and Civil Engineering Terms. For classes which have been adopted from external sources, the definition used in the source shall be applied. This is particularly relevant to the adoption of elements of STEP parts where it is important that the IFC terminology usage does not redefine the STEP terminology usage other than by allowed methods of aliasing.

Prepare Process Model

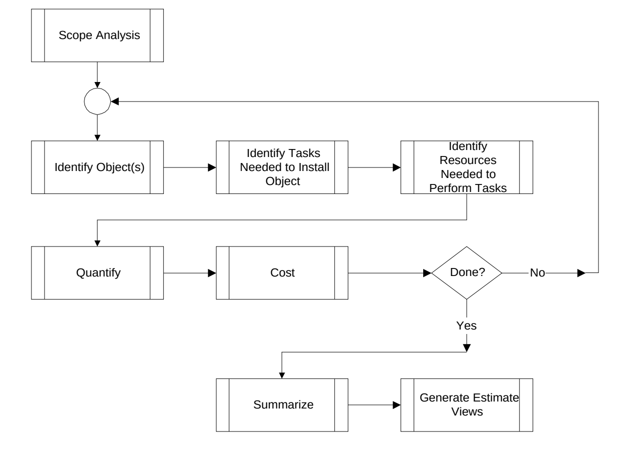


Figure 12: Example Process for Cost Estimating

A process model provides a description of the tasks needing to be undertaken to complete a defined industry process. Completion in this sense is defined by a scope statement which sets out, in broad terms, what is required of the process model and subsequent IFC specifications.

The process model defines all of the required tasks within the process and sets them into a logical sequence. Completion of the process model enables identification of the required scope of specification development.

Determine Scope

The scope statement aims at an early adjustment of necessary resources for an AEC project definition by indicating the scope of the engineering domain. It includes:

- short statement of the domain content
- short statement about what is out of scope
- short description of the intents for the development
- preliminary object list

Specify Object Model

Class Name							
	Interface name		{{ "Ref" = relationship }}				
1 2 3 4 5 6 7	***Attribute/Relation name***	***Definition***	***Data Type or Related Object Type***	***Min***	***Max***	***Default***	***Units***
2	**IfcSpaceProgramme**	Architectural programme for a space --> requirements definition					
	InheritsFrom ---------->	IfcControlObject					
	I_SpaceProgramme						
	SpaceName	Programme name for this space	IfcString	see type	see type	<Space#>	n/a
	ProgrammeType	Relationship to a SpaceStandard TypeDef object which defines the standard requirements for this type of space in the programme (NOTE: this is different than the SpaceType associated with IfcSpace)	Ref [IfcTypeDefinition]	see type	see type	NIL	n/a
	SpaceUse	Programme functionality required of this space	IfcString	see type	see type	empty string	n/a
	RequiredArea	Floor area required by the programme (may be different than that defined in the SpaceStandard)	IfcAreaMeasure	see type	see type	0	see type
(INV Rel - in interface only)	HasAdjacencyReqs	Set of relationships to Space adjacency objects (objectified relationship).	Set [0:N] Ref [IfcRelSpaceAdjacency]	0	N	empty set	n/a
	ProgrammeForSpaces	Relationship to an IfcSpace occurrence which satisfies this programme	Set [1:N] Ref [IfcSpace]	see type	see type	NIL	n/a

Synthesise Object Model

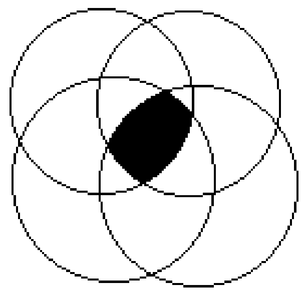


Figure 13: Union of All Specifications

Once the domain specifications are complete, they undergo synthesis. This ensures that common ideas are managed consistently and can therefore be implemented equivalently by different software applications. Synthesis takes account of:-

- Ideas already included in the IFC Object Model.
- Ideas which are common between domains.
- Ideas which are similar between domains (with a view to attempting to make them common).
- Interfaces which are common between domains.
- Interfaces which are similar between domains.
- New ideas and interfaces.

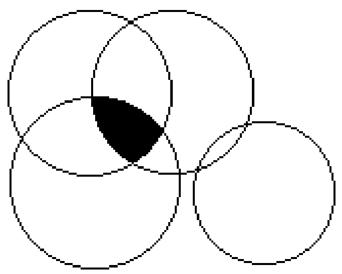


Figure 14: Union of Most Specifications

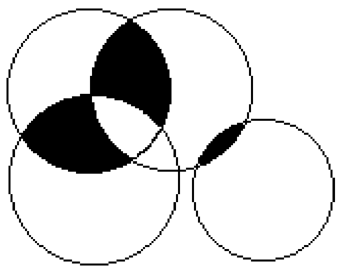


Figure 15: Common Classes Between Domain Specifications

Multifunctional Building Elements

Example

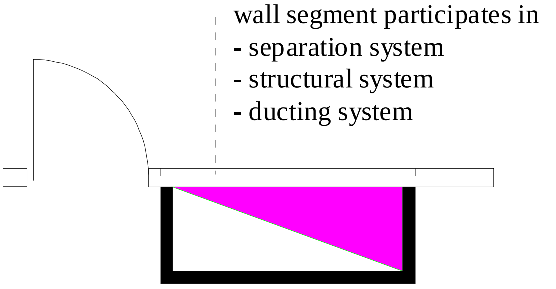


Figure 16: Wall Segment as a Multifunctional Element

Multifunctional elements are of significant interest for enabling interoperability. Many domains deal with these elements, each having its own view of the element. Domains need to identify elements of interest to them which may serve other domain functions and look to identify such elements within other domain models or, more properly, in the interoperability layer of the IFC Core Model (since this is where such concepts belong)

Changing the location of the wall segment as a result of space layout modifications may:-

- change the configuration of other wall segments which connect to that being relocated.
- require the redefinition of structural elements and will necessitate the recalculation of structural load.
- change the thermal load on spaces which the wall segment bounds.
- cause a change in the size of ductwork penetrating the wall segment.

Decomposition

The classification of decomposition strategies is an essential part of the data sheets, since decomposition brings a basic functionality into the IFC definition. The following types of decomposition can be distinguished:

I. functional decomposition
following the primary functions it fulfils in the various systems e.g., in spatial, structural, thermal system

II. constructive decomposition
following the way an element is built up according to the rules of the constructive system e.g., as layered element, as composite elements

III. geometric decomposition
following the basic shape types e.g., the cross sections or the floor layout shapes

Example of functional decomposition of wall

in structural system	in spatial system	in fire resistant system
load-bearing walls non load-bearing walls cellar walls (bear additional horiz. loads) retaining walls (only horiz. loads)	exterior walls interior walls to another building to stair case to another flat to another room	fire wall

Example of geometric decomposition of wall

Three different decomposition strategies for the geometric decomposition of walls can be described as an example:

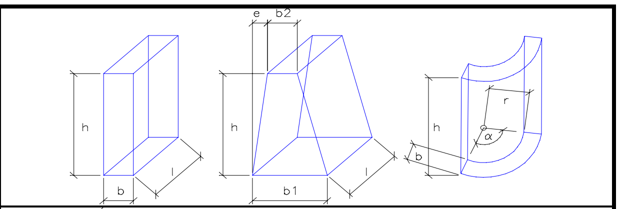
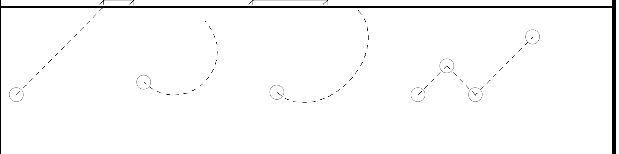
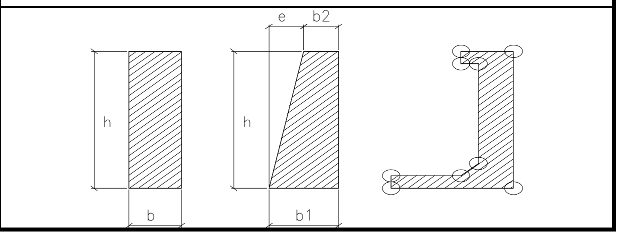
following the stereometric solid of a wall standard wall : cuboid standard conic wall : prism standard arc wall : cylindrical segment	
following the path of the wall in floor plans line arc elliptic arc	
following the vertical cross section of a wall rectangle trapezium polygon	

Figure 17: Geometric Decomposition of a Wall

Component Structure

The ability to handle whole-part relationship or component structure is another main characteristic for most building elements. There are two main strategies to handle the whole-part relation:

- aggregation / decomposition
- grouping / segmentation

Attribute Sets

For use as a checklist the following distinction between attribute sets can be made:

- common sets (used for almost all building elements)
- other, special sets (used for subsets of building elements under specific views)

- **shape parameter**
    - parameter classes for basic shape types
    - assignment of modular dimension steps
- **volume / quantities**
    - focused on quantity takeoff
    - definition of measurement rules
- **material / execution**
    - definition of material attributes
    - reference to external catalogues
- **surfaces**
    - special attribute for building elements,
    - important for the assignment to spaces

Other more specialised properties are used for a range of domain specific attributes. They can later form extended attribute sets, needed for specific tasks, such as compliance checking.

- Thermal load calculation
    - heat insulating property
    - heat transfer property
- heat protection in summer time
- moisture protection on surface
- moisture protection inside the construction
- sound protection, transmitted by air
- sound protection, transmitted by construction
- fire resistance

Relationships

Three main groups of relationships are predefined. They cover a broad range of all possible relationships. Of course, they have to be specialised, as the example of relations to other building elements shows.

I. topological relation
   A. relative positioning of elements
   B. constraint positioning of elements
II. relation to other building elements
   A. set of predefined relationships with precise meaning
   B. may be specific to domain and participating building elements
III. relation to space elements
   A. space separating relationship of bounding building elements
   B. space reduction relationship of space enclosed building elements

Example
Specific relationships between building elements:

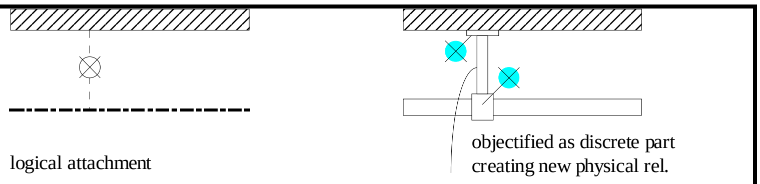
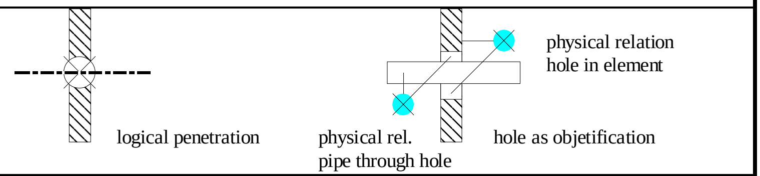
attached to	
penetrated by	

Figure 18: Specific Relationships Between Building Elements

Review Procedure

It is important that IFC developments are reviewed by industry experts and information modelers other than those who created the IFC specification for the following reasons:-

- Ensures that the specification is validated by the agreement of a wide cross section of industry experts.
- Identifies aspects of the declared domain process which have not been fully included within the specification.
- Ensures that the specification is applicable in an international context and does not relate only to the working practices of a particular region.
- Provides a wider range of expertise for the development of the specification.
- Allows for the correction of inaccuracies within the specification.

Not everyone needs to review every aspect of the specification. The following table identifies typical reviewers, what they should review and their purpose in doing so.

You are …	**You should review …**	**To determine …**
A manager (either within the AEC/FM disciplines or responsible for IT applications)	The domain specification scope statement and usage scenario	Whether this domain specification meets your business needs
An industry expert, or a modeller responsible for developing specifications and systems that support business processes	All of the above, plus The information requirements and their supporting material	Whether all the data that you use for the business process is completely and correctly identified
An applications software developer, or a modeller involved in designing and developing systems	The IFC Domain Specification and its supporting material The exchange classes	How the Domain Specification can be mapped to an application's data, or to data held in database(s) Which exchange classes are aligned with the scope of your application

A fundamental aspect of review is that it should be undertaken not only within the local Chapter in which the specification is originated but also within other Chapters so that when complete, it is a genuine representation of an international consensus.

Readers Guides

The IFC Model Development Guidelines provides Readers Guides on the process and information modelling techniques used in development. These are not detailed references but are aimed at providing a sufficient level of understanding of basic concepts so that industry experts and information modellers have a common reference point

Conclusions

It is less than two years since the formation of the original Chapter of the IAI in North America. It is now an international organisation with hundreds of members in the AEC/FM and its related software industry.

In two years, IFC development has undergone two release cycles. Within those cycles, many lessons have been learned and these are being applied.

- We have learned and applied the lessons of a flexible Architecture.
- We have learned and applied the lessons of flexible, automated and synchronised documentation.

- We have learned the need of providing guidance to domain specifiers and provided them with Guidelines.

Importantly, we have learned that none of what we do can be regarded as static. All of the lessons must be revisited periodically and the means by which we provide solutions reconsidered in the light of changing circumstances.

We have many more lessons to learn. However, the IAI has already shown itself to be capable of adapting to the challenges it faces in bringing interoperability to the AEC/FM industry and there is good reason to expect this dynamic approach to continue.

This paper represents some of the solutions adopted in the development of IFC Release 1.5. By the time you read it, we will probably already be working on enhanced solutions for versions 2.0 and beyond.