HIGHLIGHTS OF THE DEVELOPMENT PROCESS OF INDUSTRY FOUNDATION CLASSES

Development process of industry foundation classes

T. LIEBICH
Thomas Liebich Consultancy, München, Germany & AEC3 Ltd., Thatcham, UK
J. WIX
Jeffrey Wix Consultancy, Thatcham, UK & AEC3 Ltd., Thatcham, UK

Durability of Building Materials and Components 8. (1999) *Edited by M.A. Lacasse and D.J. Vanier*. Institute for Research in Construction, Ottawa ON, K1A 0R6, Canada, pp. 2758-2775.

Abstract

The Industry Foundation Classes, IFC, are developed by the International Alliance for Interoperability, IAI, to provide for data exchange and sharing capabilities for the building and construction sector of the industry. Since the foundation of the IAI in 1995 two releases of IFC have been published and the third release is on its way. Many prototype implementations by leading software companies have proved the IFC specifications, and commercial products to satisfy the end user demands can be expected within the next few quarters. The paper highlights the major development steps between the recognition of end user needs to improve business processes and the fulfillment of those needs by IFC specifications and implementations. Emphasis is put on the modeling of business processes, the provision of use cases, the integration of domain requirements within a single integrated but modular information model, and the underlying methodology and model architecture.

Keywords: Information model, Data exchange and sharing, Software Interoperability

1 Introduction

Since the IAI was established internationally in 1995 two major IFC Releases have been published:

- IFC Release 1.0 in January 1997 (IAI 97a)
- IFC Release 1.5 in November 1997 (IAI 97b), including the technical corrigenda published as IFC Release 1.5.1 in August 1998 (IAI 98)

During the 3 year period major technical goals have been achieved but some adjustments have also been made to the development methodology and architecture. This occurred concurrently with the continuous growth and international coverage of the IAI, which now has around 600 member companies within 8 international chapters covering over 20 countries.

The technical answer to the challenge of collaborative development between various contributors internationally and within several domains was the subject of two major documents in addition to the IFC model specification:

- IFC Specification Development Guide (IAI 97c)
- IFC Object Model Architecture (IAI 97d)

The authors participating in the development of all documents mentioned above want to highlight the major development tasks as defined within the Guide and to present the ideas behind establishing a well defined and rigid model architecture to maintain a single integrated but modular IFC object model.

By the time this paper is presented, the current version of IFC will be implemented as pre- and postprocessors in commercial software by leading software companies, such as by Autodesk for the Architectural Desktop, by Graphisoft for ArchiCAD and by Nemetschek for Allplan, to name a few. These companies are using experience gained by prototype implementations over the last 1 ½ years, in which over 40 companies from all over the world have been involved.

The next major deployment step will be demonstration projects where IFC based technology is successfully used within real construction projects. Ideas and first initiatives are described in (Wix & Liebich 98).

1.1 Related work

The relatively fast development of IFC from its initial concept in 1996 to commercially implemented release in 1999 would not have been possible without the experience and information sources coming from other previous or ongoing projects and initiatives.

Starting from the GARM, General AEC Reference Model (Gielingh, 88), various attempts have been made to use Product Modelling techniques to capture the semantics within the construction objects. Within the European ESPRIT projects, the projects ATLAS, COMBINE, COMBI and now VEGA, TOCEE and CONCUR explored the applicability of Product Modelling within demonstration scenarios coming from the construction industry. Other major input came from the ISO standardisation efforts in STEP, “Product data representation and exchange”. In particular the EXPRESS language (ISO10303-11 94) and the STEP physical file format (ISO10303-21 94) are used as the fundamental technologies in IFC. Moreover, parts of the Integrated Generic Resources have been adopted to form part of IFC resources in order to reuse proven concepts coming from other industry areas. The development of Part106 also influenced the IFC and vice versa (ISO10303-106 97).

Acknowledging already existing standards within the same industry area, most notable AP225 (ISO 10303-225 97) and CIMsteel (CIMsteel 94), the concept of an Interoperability Adapter has been incorporated into the IFC structure from the beginning to establish the scenario of schema mapping to external Product Models. This has been strongly facilitated by the execution of a liaison agreement and memorandum of understanding with ISO TC184/SC4 (ISO

TC184/SC4 98).

New technologies

mplemented mainly

IFC specification are more far reaching, including (see figure 1):

partial data exchange,

software interface based implementa

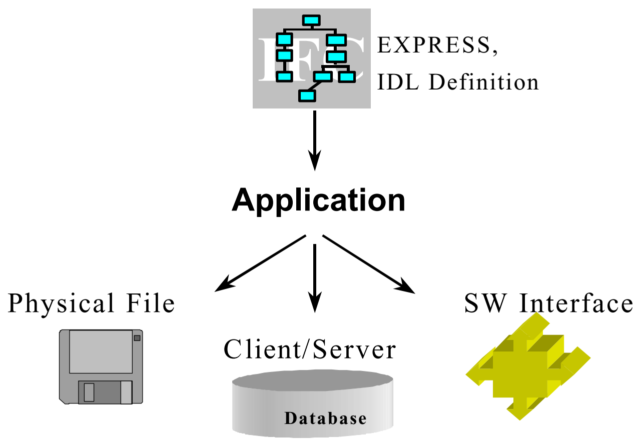


Fig. 1: Implementation scenarios for IFC

In order to support these implementation scenarios the IFC model development:

follows modular architecture approach,
restricts EXPRESS language features that would impede alternative implementation,
adds additional specifications into the IFC model documentation.

The EXPRESS specification of IFC is complemented by a prototype IDL specification, according to the Interface Definition Language, as published by the OMG (OMG 95) and ISO 10303 part 26 (ISO10303-26 98).

2 IFC development cycle

The cycle of one IFC release comprises the following mayor steps:

1. Define user requirements
2. Define business process within a project
3. Test by usage scenarios
4. Specify domain models
5. Integrate into current release of IFC
6. Review new release of IFC
7. Finalise documentation

8. Define conformance
9. Support implementation

The accomplishment of all tasks requires the active participation of many contributors. The IFC Specification Development Guide (IAI 97c) provides the modus operandi for the collaboration.

2.1 Roles within the development cycle

IFC specifications are developed by groups of experts. Each expert group performs various development steps and reviews others. The principal groups that can be identified are:

- project groups for defining the requirements;
- technical experts for specification and integration;
- software vendors for implementation.

The roles and responsibilities of the groups are presented based on the following premises:

- the knowledge required for collaboration (overlapping fields of knowledge);
- the responsibilities for the various development processes;
- the review tasks and the documentation requirements within the development processes.

2.1.1 Domain experts

A project develops industry requirements for interoperability within its area of interest. The requirements are then presented in a form that enables a formal information model to be developed.

Participants within a project include domain experts who must have technical expertise in its area of interest. It is useful if members also have knowledge of the capabilities of application software currently used within that domain. Projects are responsible for development tasks 1 to 3, as outlined in section 2. Major work to be undertaken by domain experts within a project is:

- project initiation and scope definition;
- technical analysis and usage scenario;
- specification review.

2.1.2 Technical committees

The International Technical Management Committee (ITM) is the principal technical body within the IAI. All other technical committees report to the ITM. Its responsibilities are to:

- provide the technical planning and management required for IFC development;
- co-ordinate international technical work;
- monitor the IFC specification process.

ITM is responsible for the acceptance of AEC project proposals (result of development tasks 1 to 3) and for the acceptance of the final IFC Release (result of tasks 4 to 7).

The Specification Task Force (STF) provides high level technical support to IFC specification developers and implementers, to guide the object model specification work and to produce the necessary technical documentation. Within the STF, technical experts perform product definition work and are responsible for IFC model synthesis. The definition and maintenance of the IFC model architecture belongs to their tasks as well. STF is responsible for development tasks 4,5,7 and provides heavy input into 8 & 9. The following tasks are the responsibility of the STF:

- develop the technical architecture;
- define or supervise the specification of domain models coming from AEC project proposals;
- integrate IFC definitions;
- maintain previous IFC releases;
- continuously enhance the fundamentals of the IFC specifications;
- develop the IFC Release document suite;
- support software implementers.

2.1.3 Software implementers

The Software Implementation Committee allows software implementers to work together in a spirit of open, pre-competitive collaboration to ensure that the IFC Object Model can be implemented in practice. Software implementers are responsible for:

- pilot implementation of selected IFC data or interface definitions;
- realisation of commercial implementation according to the companies' business plans;
- feedback into the specification work;
- review of IFC data or interface definitions and project definitions.

Software implementers have a particular responsibility to the IAI in providing both the pilot and commercial implementations of IFC. It is only through such implementations and the demonstration of their capability that IFC will be deployed throughout industry and that support of the IAI by industry will grow.

2.2 Development tasks

This section presents an overview of all tasks necessary to develop solutions for current gaps in information sharing as indicated by professionals. An example coming from engineering maintenance is used throughout the section.

All developments should start with the recognition of potential business benefits by professionals in industry. To illustrate the potential improvements expected from proper information handling in engineering maintenance the following assessment had been made by a senior bank representative: 25% of the total facilities maintenance budget deals with engineering maintenance, if 10% of that could be saved, the total savings would be in a range of $15m to $30m for the properties owned by a major bank (Goodman 97, see also Wix & Liebich 98).

2.2.1 Definition of user requirements

This first stage is about deciding the project on which work is to be carried out. The objective is to select one or more industry processes that can be broken down into manageable and useful parts. Each part should be able to be completed within the time scale of the target IFC release.

Once the industry processes has been selected each task within the process needs to be described. The description follows the wording as normally used by professionals and is complemented by figures (example given in Fig. 2 and Table 1).

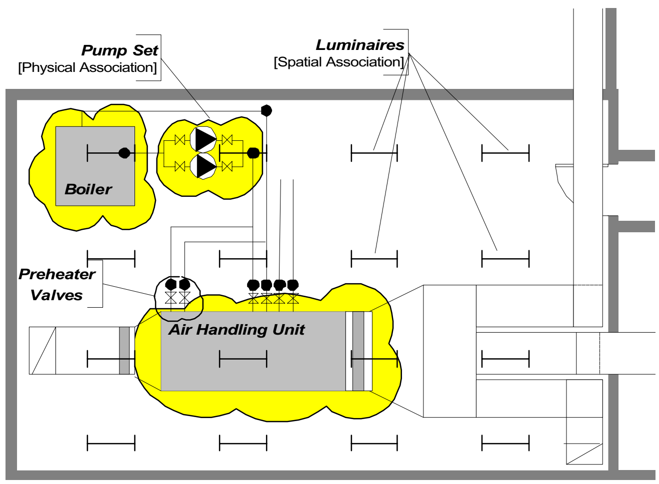


Fig. 2: Task: Group objects as assets, shown as figure

Table 1: Task: Group objects as assets, shown as text box

Example task description **Group objects as assets** ▪ This task involves the specification of the asset. An asset can be either a single object or may be a group of objects upon which a single maintenance action is to be specified. ▪ A large item of equipment such as an air handling unit, boiler or transformer will probably be uniquely identified as an asset. ▪ An asset may be declared as a group of objects with a spatial identity such as the light fittings within a room. ▪ An asset may be declared as a group of objects with a system identity such as the steam traps within a system (or subsystem).

2.2.2 Definition of domain processes

A process model is a formal description of the tasks undertaken to complete a defined industry process and sets them into a sequence. For the next stages it provides the basis for identification of the object model scope.

Process models are fairly easy to understand. Domain experts, with some guidance from modeling specialists, can create and understand process models with little training. A process model contains:

- tasks performed during the process execution;
- decomposition of processes in a hierarchy;
- sub-processes that are usually nested within processes;
- relations between processes, forming a sequence;
- main relations are input, output and control;
- mechanisms may also be defined which provide the means of executing individual tasks.

The process modeling method used for IFC development is an adapted form of TQM modeling, an example is given in Fig. 3. The use of IDEF0, the most recognized process modeling language, is proposed for the future (IDEF0 93).

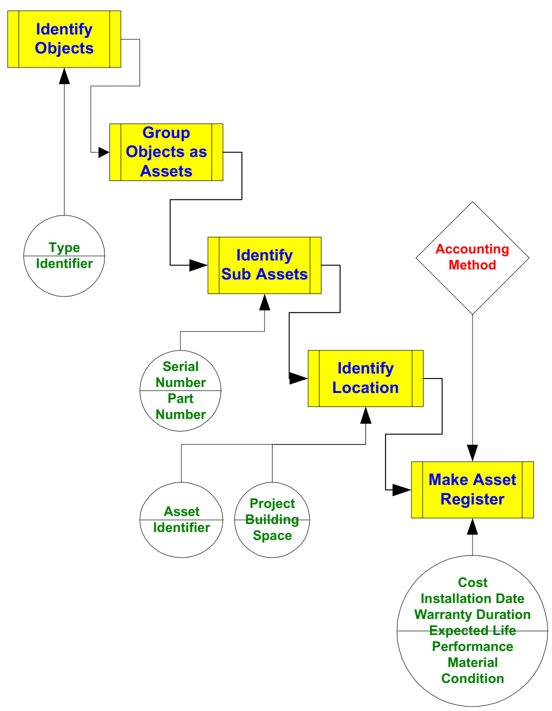


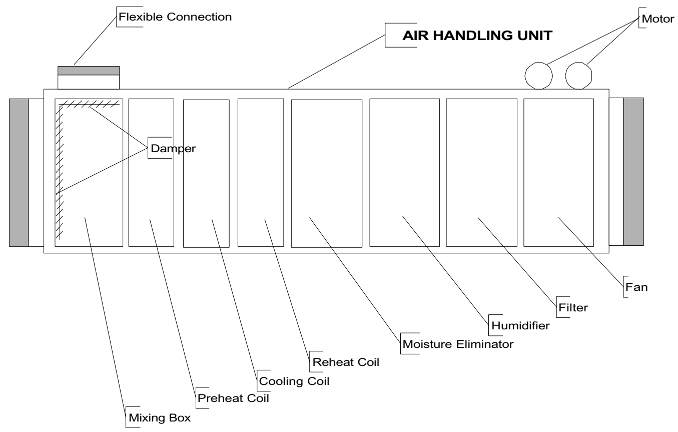
Fig. 3: Process model for sub process: Asset register

The usage scenario is a textual description that sets down the complete information requirement in such a manner that it relates together the scope definition, the process model and the object specification. At this stage, the specification does not exist. The usage scenario identifies its elements (see Fig, 4 for graphical description and Table 2 for lexical description).

When creating the usage scenario, the following rules shall be observed:

1. *Be assertive*: a usage scenario should provide a set of assertions that can be modelled and implemented: An asset is located within a space;
2. *Be clear:* about the ideas being captured by the usage requirement. In the above, asset and space 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. The ‘is located on’ reads inverse as a space locates ‘zero-to-many’ assets.



Fig, 4: Air handling unit as usage scenario

Table 2: Example usage scenario

Identify sub assets

The decomposition of the asset into component parts or sub assets may be already known by virtue of the separate existence of the elements making up the asset (as in the case of the pump set and the luminaries). Alternatively, it may be necessary to identify the parts of the asset requiring individual maintenance actions where the element forming the asset is bought in as a single item. This will be the case for the air handling unit and the boiler.

Separate components of an air handling unit include:

- Fan (motor[s], impeller, drive wheels and belts)
- Filter
- Pre-heater coil
- Re-heater coil
- Cooling coil
- Dampers (fresh air and re-circulation)
- Moisture eliminator
- Humidifier

Separate components of a boiler include:

- Burner assembly
- Controls
- Heat exchanger

Drawings and sketches improve the quality and completeness of a usage scenario. They also help modelers and software implementers to understand how to realize the classes and relationships as expressed. An early indication of the data operated on in each usage scenario is required as well (see Fig. 5).

2.2.3 Specification of domain model

The domain model specification is carried out in two steps:

1. Set of assertions about statements in task descriptions, process models and usage scenarios
2. Formal data model for all classes and relationships handled by the processes in scope.

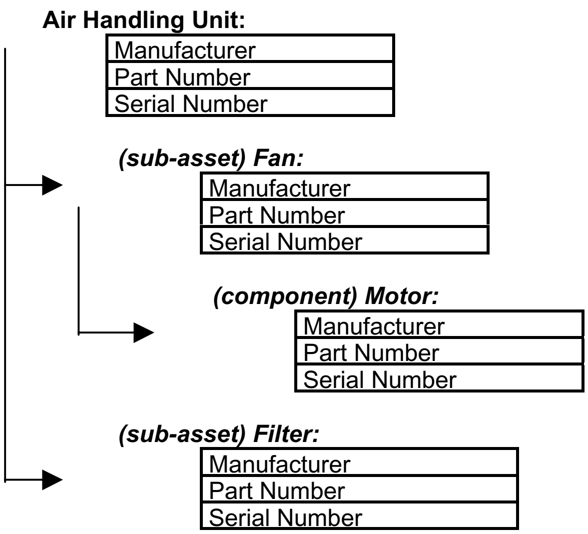


Fig. 5: (Partial) decomposition of air handling unit

A set of assertions breaks down the usage scenario into simple sentences. Each assertion is the simplest sentence that can be developed. Optimally it contains two nouns and a verb, plus some qualifying grammar that is used to describe cardinality and rules (see Table 3). The objective of developing assertions is to:

- identify classes;
- identify relationships;
- identify cardinality of relationships;
- identify direction of relationships;
- identify attributes;
- identify rules (uniqueness, derived values, constraints).

Table 3: Example assertions

Identify sub assets

- An **air handling unit** *consists of* <u>one or many</u> **fans, filters, coils, dampers, eliminators,** and **humidifiers**
- A **fan** *consists of* <u>one or many</u> **motors, impellers, drive wheels and belts**
- A **fan** *is characterised by* ***Manufacturer name, Part number and Serial number***

Class | *Relationship* | <u>Cardinality</u> | ***Attribute***

During the formal definition of the domain model the assertions are transformed into statements using an acknowledged modeling language, here EXPRESS, and the graphical notation EXPRESS-G (see Fig. 6).

The formal definition also marks the point where the transition from the responsibility of the domain team to the technical team occurs. The basic indications shall be given by the domain team suggesting which information requirement can be best realized from the user perspective as class, attribute, property set, and which are the appropriate relationships, including inheritance, aggregation, or has relation (see figure 7). Further refinement is than needed from technical and modeling experts to guarantee that the final domain model also meets the requirements of conceptual clarity and model consistency and that it follows a common modeling philosophy. All of the above ease the later task of model integration.

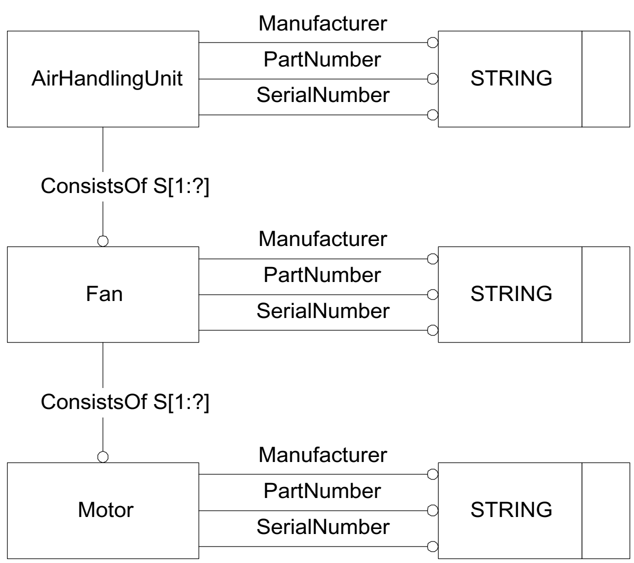


Fig. 6: EXPRESS definition example from domain model reference to this

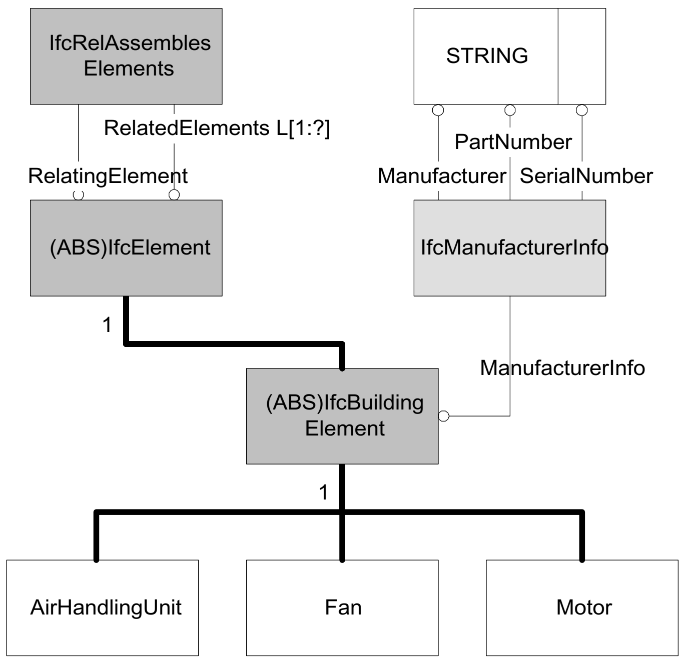
2.2.4 Integration into current release of IFC

When the domain specifications are completed, they undergo synthesis. This ensures that common ideas are managed consistently and can be implemented equivalently by different software applications. The result of synthesis is a new release of the integrated information model, IFC. Good domain specifications have already been developed by taking into account the need for synthesis by:

- using facilities provided by the existing IFC resource models;
- referencing the basic concepts already provided by the IFC core model;
- cross checking with other either already existing or newly proposed domain models for overlap.

During the integration process the modeling experts, namely the STF, carry out processes of synthesis that will take account of: (see Fig. 7)

- ideas already included in the IFC Object Model;
- ideas which are common between domains;
- ideas which are similar between domains (trying to make them into common ideas);
- truly new ideas.



Grey box : Idea already included in IFC
Grey stripes: Ideas common between domains
White box: New ideas

Fig. 7: EXPRESS definition after synthesis

The synthesis results in careful additions to the current IFC release which reflect the domain needs not already handled by existing model components. The main risk of a growing (and thereby changing) information model is, however, that previously supported definitions could inadvertently be affected by changes due to the highly interconnected nature of considerably sized information models. Two principle solutions are discussed to solve that problem:

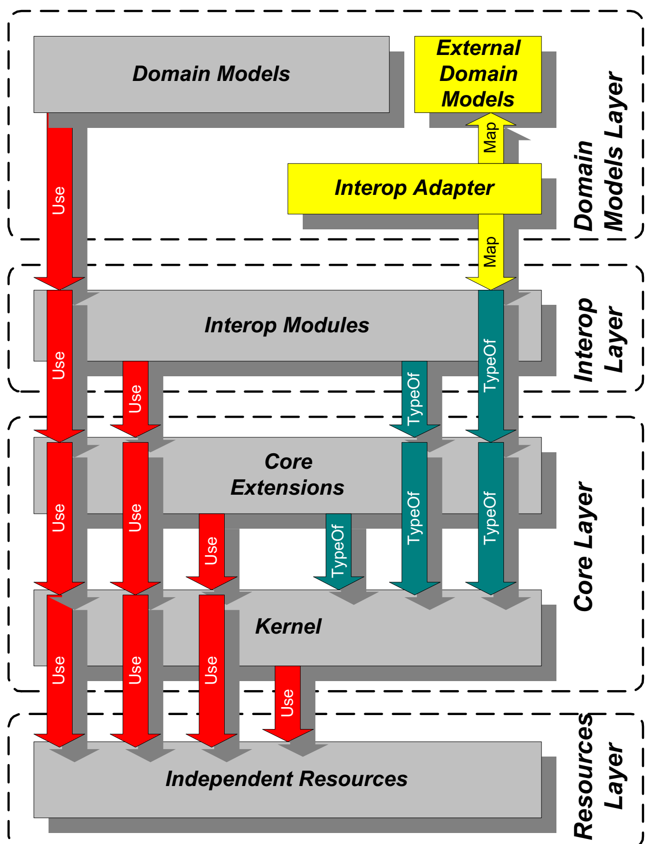
1. Prohibit any changes to the existing generic information model.
2. Provide a modular architecture for the information model to maintain changes as needed.

The STEP standardisation follows the first solution. New requirements coming from domain models (called Application Reference Model, ARM) have to be mapped down to the Integrated Generic Resources. The mapping is described in mapping tables that form part of the standard. Definitions given in the Integrated Generic Resources can only be further constrained, i.e. by providing constraints in subtypes or rules, however they can not be enhanced. The result is an Application Integrated Model, AIM. The disadvantage is, that the semantics of the domain model get confused by mapping concise domain constructs into various very generic resource definitions.

The IFC approach is different and follows the second solution. The IFC model is decomposed into many modules defined at various layers of abstraction. Strict referencing rules apply to necessary links between the modules at each layer. The architecture operates on a 'ladder principle', also known as the software engineering layering principle:

- Any class may reference a class at the same or lower layer but may not reference a class from a higher layer.
- References within same layer must be designed very carefully and are only allowed within the Core layer and the Resource layer.

Fig. 8 shows the strict downward reference within the IFC Architecture. It is described in further detail within the IFC Architecture document (IAI 97d). The IFC Release 2.0 being under development relies on the same architectural structure, but added more content to each layer.



Red arrows: Reference for uni-directional association
Blue arrows: Reference for inheritance relationship
Yellow arrows: Mapping between IFC and external models

Fig. 8: IFC Architecture reference to this?

Whereas the resource and core layers remain quite stable the incremental growth will occur within the interoperability and domain layers. Release 2.0 particularly enhance the IfcSharedBldgElements and IfcSharedBldgServiceElements for building elements shared by many domains or application types and adds the IfcSharedSpatialElements for shared space related information. The IfcArchitectureDomain, IfcHvacDomain and IfcFacilitiesMgmtDomain increases for building elements dominantly used by a single domain or application type and further domain schemas are added to deal with cost estimating and construction management.

The STF works on three versions for each release reflecting different states of the synthesis:

- *Alpha version* represents the consolidated domain models where all ideas already included in the IFC model are filtered out and the new requirements are highlighted as deltas. The alpha models are pre-integrated.
- *Beta version* represents the integrated IFC model, where the deltas from the alpha version had been added and the ideas which are found common or similar between domain models had been generalized. The beta model is the next IFC release for review and comments.
- *Final version* is the next official IFC release for implementation after all issues coming from the reviews had either been incorporated, postponed (for inclusion into further releases), or rejected

2.2.5 Review of new release of IFC

It is important that IFC developments are reviewed by industry experts, information modelers and software engineers for the following reasons:

Table 4: Review types and participants

You are …	**You should review**	**To determine …**
A manager (either within AEC/FM or 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 domain specification and its supporting material	How the domain specification can be mapped to your application's data, or to the data held in your database(s)
An applications software developer	The exchange sets	Which exchange sets are aligned with the scope of your application

- ensures that the specification is validated by the agreement of a number of industry experts;
- identifies aspects of the domain process which have not been fully included;
- ensures that the specification is internationally applicable;
- provides a wider range of expertise for the specification development and allows for the correction of inaccuracies within the specification.

Table 4 identifies typical reviewers, what they should review and their purpose in doing so. Any specialist in any of the above mentioned professions is invited to review the IFC in total or in parts.

All issues raised against an IFC model are captured within the IFC Issue Resolution Database, IRD. The STF receives all issues and decides about the status, accepted or rejected, and if accepted, whether it can be resolved within the current release or whether it has to be postponed for the next release. For each accepted issue the proposed resolution, if given, is considered and actions are assigned to the responsible information modellers within the team to finally adjust the IFC model. Table 5 shows the record of an issue.

Table 5: Printout from the issue resolution database

Issue #r	***I 276***	***Issue Date***	18-Sep-97
Author	T. Liebich	***Owner***	J. Wix
Status	Resolved		
Schema	IfcDocumentExt	***Version***	Beta
Issue Description	Class IfcRelGroupsCostSchedules The objectified relationship subtype does not define further information		
Proposed Solution	Delete and use IfcRelGroups instead; with the "GroupPurpose" set to "GroupsCostSchedules" - update documentation to make the usage clear		
Resolution	agreed		
Action #1			
Assignee	J. Wix	***Status***	Complete
Resolved in Version	1.5 Final		

2.2.6 Final documentation

The final version of an IFC Release is published by a series of documents:

- Introduction to IAI and the IFCs
- IFC Object Model Architecture Guide
- IFC Specifications Development Guide
- IFC Object Model Specification
- Vol. 1 – AEC Processes supported by IFC
- Vol. 2 – IFC Object Model Guide
- Vol. 3 – IFC Object Model Reference
- Vol. 4 – IFC SW Certification Guide

Whereas the IFC Object Model Specification is only available to the IAI membership, the other documents are in the public domain. To ease the access to the model specification, the Introduction to IAI and the IFCs and the IFC Object Model Reference are also provided as HTML documents. The IFC Object Model Reference is fully hot linked so that all parts (the specification text, the EXPRESS definition, the IDL definition, and the Property Set definitions) are inter linked (see figure 10).

The other two steps, as mentioned in section 2, Conformance class definition and Implementation support, are not discussed within this paper. Both are of particular interest to the IAI software developer community.

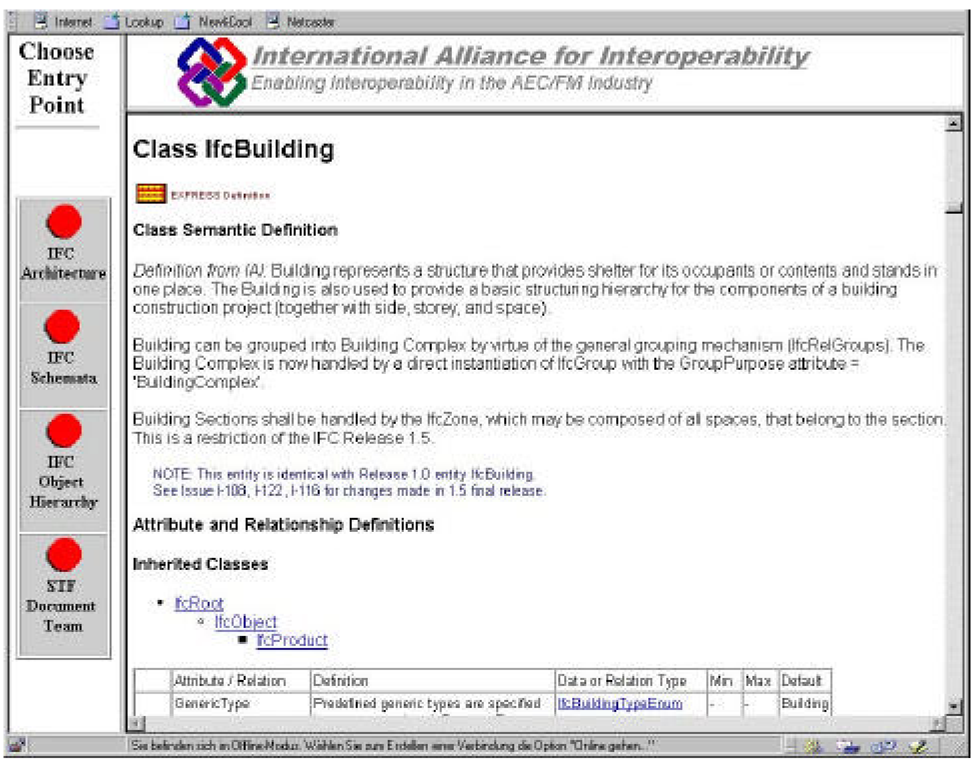


Fig. 9: Snapshot of the HTML documentation of Volume 3

3 Conclusion

None of the individual steps as outlined above are new; they all had already been described in various papers and had been used in several projects.

What is remarkable, however, is that within the IFC development process they had been merged to a methodology applicable for industry-wide and commercial use. Even as this paper is presented, leading software companies are implementing **and** deploying IFCs within their commercial products. The companies involved have a market share of 2/3 of the European market for architectural software.

4 Addendum

This paper refers to the development of IFC Release 1.5.1. Presentation to the CIB W78 meeting will include example material taken from IFC Release 2.0 which is scheduled for public issue in April 1999. Release 2.0 provides specifications that meet the business process requirements of the processes as shown in Table 6. The following picture shows the intermediate stage of the IFC Release 2.0 architecture that reflects the growth in comparison with IFC Release 1.5 (see also Fig. 8.).

Table 6: Domain projects for IFC release 2.0

Architecture AR-1 Enhancement of the Architectural Model AR-2 Space planning for escape routes	Facilities Management FM-3 Property Management FM-4 Occupancy Planning
Building Service BS-1 HVAC Systems design BS-3 Pathway co-ordination with other trades BS-4 Thermal loads calculation	Codes and Standards CS-1 Code compliance checking CS-2 Code extensions
Simulation SI-1 Photometrically Accurate Visualisation	Estimating and Scheduling ES-1 Cost Estimating
Cross Domain XM-3 Network Model and Model Constraints	

5 Acknowledgement

The authors want to acknowledge the consensus driven team effort within the Specification Task Force that enabled the realisation of IFC. The IFC Release 1.5.1 was developed by James Forester, Richard Junge, Jean-Luc Monceyron, Richard See, Kevin Yu, and the authors. For IFC Release 2.0, the team has been strengthened with the addition of Juha Hyvarinen, Kari Karstila, Robin Drogemuller and Jiri Hietanen

The examples have been taken from the IFC Release 3.0 domain project FM-1 Engineering Maintenance developed under the leadership of Mike Goodman with technical support by Jeffrey Wix.

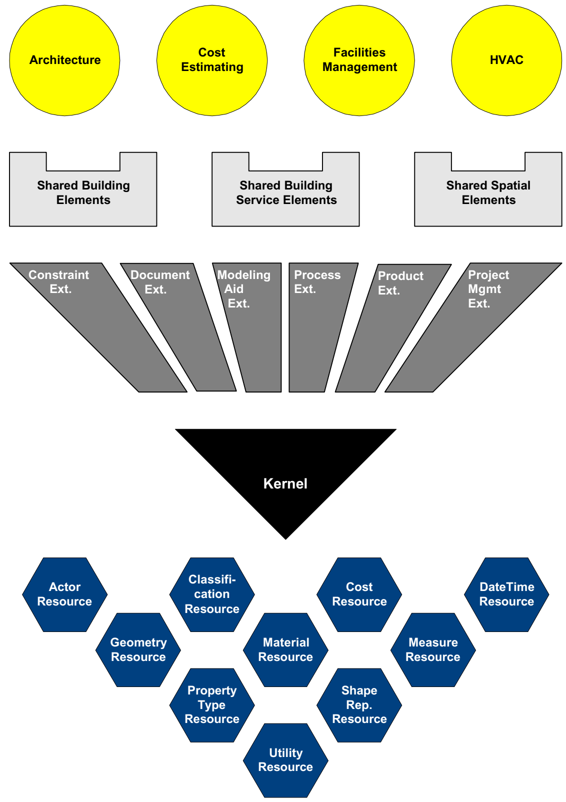


Fig. 10: IFC Release 2.0 Architecture

6 References

CIMsteel 94. Definition of a Products Library for Steel Structures, Design Task Group

Gielingh, 88. General AEC Reference Model, ISO TC 184/SC4/WG1 DOC N 3.2.2.1

Goodman 97. Paper to IAI UK Launch of IFC Release 1.5., Nov 16th 97

IAI 97a. Industry Foundation Classes, Release 1.0, IFC Object Model For AEC Projects, International Alliance for Interoperability, Jan 1997

IAI 97b. Industry Foundation Classes, Release 1.5, IFC Object Model Reference, International Alliance for Interoperability, Nov 1997

IAI 97c. Industry Foundation Classes, Release 1.5, IFC Specification Development Guide, International Alliance for Interoperability, Nov 1997

IAI 97d. Industry Foundation Classes, Release 1.5, IFC Object Model Architecture, International Alliance for Interoperability, Nov 1997

IAI 98. Industry Foundation Classes, Release 1.5.1, IFC Object Model Reference, International Alliance for Interoperability, Oct 1998

IDEF0 93. Integration Definition for Functional Modelling (IDEF0), US National Institute of Standards and Technology, FIBS Pub 183

ISO10303-11 94. Description methods: the EXPRESS language reference, Industrial automation systems and integration: Product data representation and exchange Part 11, Intern Standard

ISO10303-21 94 Implementation methods: Clear text encoding of the exchange, Industrial automation systems and integration: Product data representation and exchange Part 21, Intern Standard

ISO10303-26 98 Implementation methods: IDL Binding, Industrial automation systems and integration: Product data representation and exchange Part 26, working draft

ISO10303-106 96 Integrated application resources: Building Construction Core Model, Industrial automation systems and integration: Product data representation and exchange Part 106, working draft WG3 N599

ISO10303-225 97 Application protocol: Building elements using explicit shape representation, Industrial automation systems and integration: Product data representation and exchange Part 225, Draft Intern Standard WG3 N510 (T12)

ISO TC184/SC4 98 Memorandum of Understanding between ISO TC184/SC4 and the International Alliance for Interoperability, February 1998

OMG 95. The Common Object Request Broker: Architecture and services specification, revised edition, Object Management Group

Wix & Liebich 98. Industry Foundation Classes: Some Business Questions Examined, Proceedings of the 2nd EC-PPM conference, Watford, UK Oct 19.-21. 98