

# **Integrating the IFC Building Product Model with Zone Fire Simulation Software**

Spearpoint M J

*Department of Civil Engineering, University of Canterbury, Christchurch, New Zealand*

There is an international effort to develop a standardised, object-oriented “building product model” that overcomes the restrictions of conventional electronic descriptions of buildings and similar facilities. This product model is known as the Industry Foundation Class (or IFC) Model. The IFC Model consists of entities that represent a building; the properties needed to describe those entities and the inter-relationship between entities.

This paper briefly describes the structure and content of the IFC building product model and identifies those parts of the model that are relevant to fire engineering. The ability of zone fire simulation software to interpret IFC files is discussed and this is demonstrated using the CFAST and BRANZFIRE software. The interpretation of an IFC file is limited by the capabilities and restrictions of the fire simulation software as well as the IFC model.

## **1. Introduction**

Fire simulation software (commonly referred to as ‘fire models’) are often used by fire engineers to develop and analyse design scenarios. In order to carry out the design process, a considerable amount of time can be spent specifying the input to the simulation software prior to execution. This information can be quite voluminous and needs to be accurately transferred to the simulation software. Items such as the geometry and topology of a building, the location of items within the building and the properties of those items are just some of the information that may be required. Mistakes and missing information when providing the input information can lead to inappropriate output from the simulation software.

Traditionally, such information might be entered by hand although the building plans may have been generated by Computer Aided Design (CAD) software. Even where there is the ability to directly transfer CAD data into fire simulation software there can be significant limitations. These limitations stem from a variety of reasons including where the fire simulation software tool may be incapable of interpreting any kind of CAD file or the CAD file does not adequately describe the building in a form that the software can extract the relevant details.

Ideally we would like to have a method that allows us to specify a building in a single format and then be able to transfer that description to any software tool we require with the maximum utilisation of the initial description and the minimum need to add additional information. Furthermore, we might want to append new information to the initial description of the building as a result of performing a simulation. Mowrer [1] proposed the integration of CAD information with fire specific properties of a building and its contents. The key features identified by Mowrer required by CAD systems to permit integration included object-orientation, the association of attributes with objects and the ability to extract attributes from a CAD-developed drawing database.

## **2. Zone Fire Simulation Software**

It is not the purpose of this paper to provide a detailed description of zone fire simulation software or their underlying models. Quintiere [2] provides a description of the basic conservation equations and relationships that are used by most zone fire simulation software. The atmosphere within a compartment is normally split into two zones; the hot gas layer as result of the fire and the cool layer below. The physical conditions within these layers are considered vertically and horizontally uniform. A survey [3] has shown that there are many such fire simulation software tools available. The software ranges in complexity in terms of the input requirements, the output capabilities and the extent of a fire hazard scenario represented.

Zone fire simulation software tools often make a number of assumptions or simplifications regarding the geometry of the spaces being modelled and the materials involved. For example, it is typical but not necessarily always the case to assume that spaces have a rectangular foot-print and that ceilings are smooth and horizontal. Assumptions such as these will need to be accounted for as we attempt to map the electronic description of a real building to the idealised view required by some zone fire simulation software. Some zone fire simulation software

tools have a limited number of allowable spaces and/or a limited number of connections such as doors between spaces. In the zone fire simulation software these connections might be assumed to be in the centre of the parent wall rather than at its actual location.

Although zone fire simulation tools all follow the same basic philosophy regarding the way in which the fire environment is represented, individual fire zone simulation tools may have components that are not present in other tools. For example, one tool may have a glass breaking algorithm which requires specific information not required by other tools that do not implement such an algorithm.

### **3. Product models**

In general, any product can be considered to consist of a collection of *entities*. A *product model* expresses the type of entities that represent the product; the properties that are needed to describe those entities and the inter-relationship between entities. The description of a product model is commonly known as its *schema*. A *building product model* is a product model that specifically relates to buildings where entities may be physical objects such as doors, windows, walls etc. or more conceptual entities such as spaces or processes, contractual details etc. So for example, within a building product model a door entity has specific properties (such as its dimensions and construction material/s) and the relationship with the wall in which it is located (its position, orientation etc.).

Fire engineering is only one of the domains in the Architectural Engineering and Construction / Facilities Management (AEC/FM) industry that can benefit from a building product model [4]. Other domains might include architecture, structural engineering, environmental engineering, building services and many others. Many parameters related to a building are common to a range of disciplines including fire engineering. These parameters may include the building geometry and topology, the materials and components used in the construction and the location of the structure within the broad environment. However, due to the specialised needs of fire engineering, there are also parameters that are unique to the domain.

There are a number of issues to consider when examining the ability to describe a building in a product model and transfer that description to a simulation software tool. We will find that all of the issues discussed in the paragraphs below related to the specification, implementation and interpretation of the IFC Model impact on the integration with zone fire simulation software.

#### **3.1 Scope of the product model**

Product models can be thought of being of two types; either they are general or they are domain specific [5]. A general product model supports the generation and sharing of project data through the complete building lifecycle amongst a diverse range of domains. A general product model does not attempt to include every aspect of a product as this would likely be too complex and take too long to develop. Instead a general product model describes entity types at relatively high level.

Conversely, domain specific models retain as much of the project data as required for use within a domain. For the fire engineering domain we might find that a general product model does not provide all of the detail we need for fire simulation software tools or any other fire engineering related task. A domain specific product model could be created which has all those entities relevant to the fire engineering domain but this can lead to problems when sharing that project data with participants in other domains. There would need to be a number of software tools available to interpret each domain specific model. Furthermore, domains outside fire engineering might find the domain specific model has essential information missing or not in a form that is useable by them.

#### **3.2 Implementation of the product model**

Even where a product model completely describes a domain, we might find that specific software tools may only implement a reduced proportion of the whole product model. This can mean that although the product model has an entity described within its schema, the software tool cannot be used to create a specific instance of that entity or completely populate the properties associated with the entity. In some cases it may be possible to manually add entities in lieu of having an appropriate software tool.

#### **3.3 Extraction of domain specific information**

Finally the interpretation of the product model may lead to problems. The structure of entities may not be compatible with the specific requirements of the target software tool. This can happen where there might be insufficient detail in the product model as discussed above but also where the requirements of the software tool

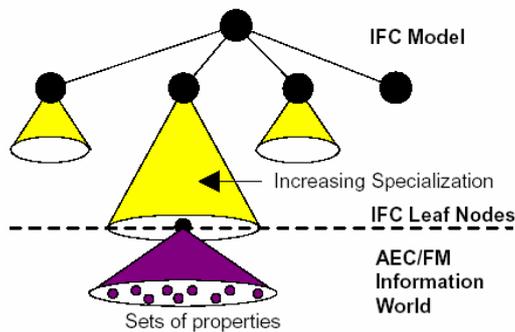
include simplifications and assumptions about a product that need to be accounted for during the transformation process.

### 3.4 Static and dynamic views

The IFC Model only provides a static view of a building whereas we may be interested in a dynamic system when we consider our fire simulation requirements. For example, we may wish to change the opening proportion of a door during the scenario or alter the flow paths along ducts during the operation of a smoke control system. In its current state, the IFC Model is unable to provide this type of dynamic information and the user would need to modify their input before executing the simulation software.

## 4. IFC Model

The IFC Model development began around 1996 and is an extension of a number of earlier projects. The latest version of the IFC Model is 2x [6] which builds on the earlier 1.5 and 2.0 major releases of the model. In November 2002 the IFC 2x Model was accepted by the International Standards Organisation (ISO) as a Publicly Available Specification, ISO/PAS 16793.



**Figure 1.** Limits of the IFC Model from Liebich & Wix [6].

Primarily IFC files are exchanged using STEP (Standard for the Exchange of Product model data) technology, ISO 10303 [7] however more recently an equivalent format using XML (eXtensible Markup Language) has been made available referred to as 'ifcXML' [8].

The IFC Model addresses the limited scope of a general product model using *property set definitions* (PSD). The high level entities terminate at "leaf nodes" which allow object types to be extended using the PSD sub-schema (Figure 1). The specification of PSDs can be made outside of the main IFC Model by specialists within

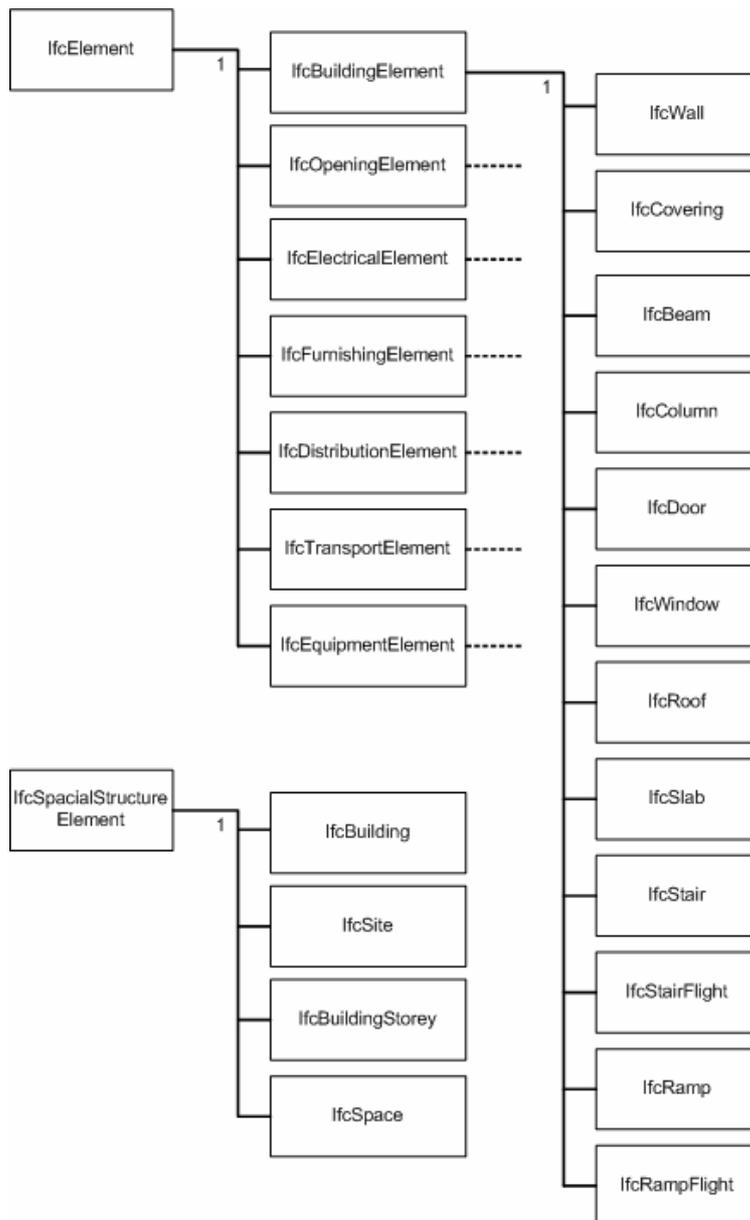
their domain. Even with the use of PSDs, there may still be specific object types missing or incomplete simply due to the fact that nobody has yet included the information in the product model. This is the case with many of the entities we might want to use as fire engineers. A review of the properties in the current IFC Model [9] with respect to fire engineering found that whilst there are a number of fundamental material properties and several regulatory related properties in the IFC Model, there are also many areas in which the model can be extended.

Even in its current 'limited' form, the IFC Model is very complex with over 350 individual entities specified in the schema. In many cases building elements are described by a generic entity with specific elements defined as sub-types of that generic entity. For example, *IfcElement* is used as a generic entity of all components that make up an AEC product of which *IfcBuildingElement* is one type. The *IfcBuildingElement* entity is used to describe specific building elements such as walls, windows, stairs etc. using *IfcWall*, *IfcWindow* and *IfcStair* entities (Figure 2). Other sub-types of *IfcElement* such as *IfcOpeningElement* also have one or more sub-types.

Although IFC specifications had been available since late 1996, implementation activity was very limited because most companies were waiting to see what the large CAD vendors would implement. A number of organisations formed the BLIS (Building Lifecycle Interoperable Software) project [10] so that actual applications could be developed and demonstrated. Software tools included CAD, thermal design, quantity take-off, model consistency checker and others. The BLIS project used the then current IFC Release 2 model as its basis.

## 5. An ifcXML Parser

A general set of classes has been written in C++ in order to extract information relevant to fire simulation software tools. These classes have then been used to create an ifcXML Parser program. These classes were developed independent of any specific fire simulation tool, be that a zone, Computational Fluids Dynamics (CFD) or an evacuation simulation tool etc. The classes interpret the IFC Model into an intermediate set of data structures relevant to fire simulation software. Interfaces to specific models can then be written to translate the intermediate data structures into a format compatible with the appropriate fire simulation software. Finally the ifcXML Parser software is provided with a Windows-based user interface for ease of use (Figure 3).



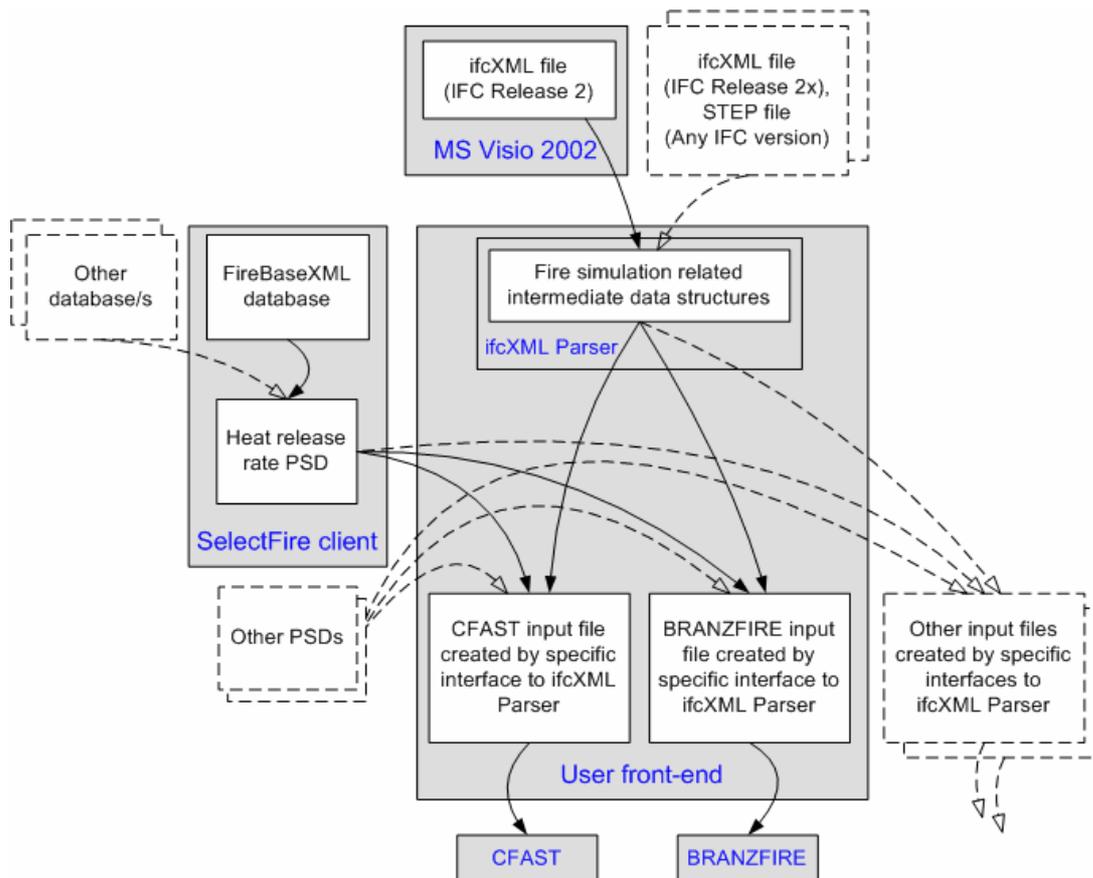
**Figure 2.** Examples of the relationship between generic and specific entities in the IFC Model.

Advantages of this approach are that should the IFC Model change then only the parser classes need be modified to reflect those changes without having to rewrite the interfaces to the intermediate data structures. Additional interfaces can be easily developed using the parser classes as their basis. The fire simulation specific interfaces can also use PSD data supplied from an external resource to enhance the translation output as demonstrated in this paper.

## 6. Software Tools

Two widely available multi-room fire zone simulation tools have been selected to demonstrate the integration with the IFC Model. The Fire And Smoke Transport (FAST) family [11] of zone fire simulation software (i.e. HAZARDI, FAST, CFAST, FASTLite) has been available from NIST for well over two decades. Many fire engineers are familiar with these software packages and they have been extensively used for a wide variety of analyses [12]. The BRANZFIRE zone fire simulation software [13] is available from the Building Research Association of New Zealand (BRANZ) and is under continued development.

Microsoft Visio Professional 2002 was chosen to create building descriptions and export them as IFC files. MS Visio is an additional part of the MS Office suite and an IFC “addin” is freely available from Microsoft. MS Visio can only exchange IFC files in XML format and the IFC addin was one of the tools developed as part of the BLIS projects. MS Visio is a 2D draughting design tool which includes templates for building core; wall, shell and structure and many other related elements. Other mainstream CAD software (such as ArchiCAD, AutoCAD, Microstation) have or are likely to have the ability to use IFC files.



**Figure 3.** Data transfer between an IFC Model file and fire simulation software (dotted items not currently implemented).

## 7. Test Buildings

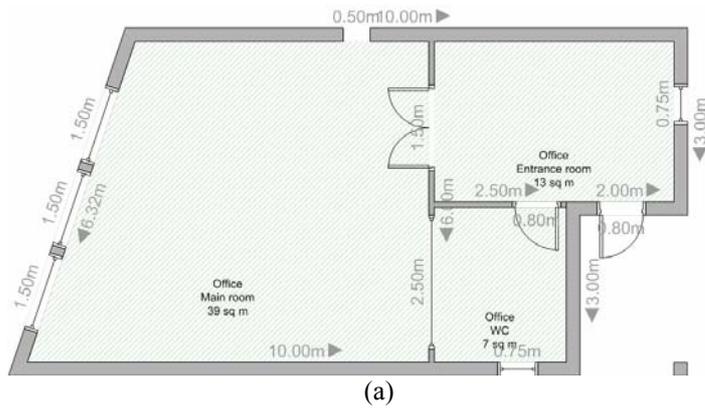
Due to the complexity of the IFC Model it was impractical to develop the ifcXML Parser without using test case buildings as a means of verifying the effectiveness of the translation process. Primarily two relatively simple buildings were used as these test cases. The basic information we want to transfer from an IFC Model compatible CAD plan is the geometry of the spaces, connections between spaces and the material properties such items as wall linings etc. Both of these buildings highlighted a number of the issues related to the specification, implementation and interpretation of the IFC Model.

The first building was a 3-room, single storey structure that is available from the BLIS website ([http://www.blis-project.org/BLIS\\_XML/simple2\\_001204.zip](http://www.blis-project.org/BLIS_XML/simple2_001204.zip)) as shown in Figure 4(a). The second test case was the ground floor of a two-storey family home was directly draughted in MS Visio shown in Figure 4(b). The building geometry was based on a house which has been used in a number of fire-related experimental projects [14]. Several furniture items, taken directly from the MS Visio furniture types, were arbitrarily added to the house in order to test some components of the ifcXML Parser software.

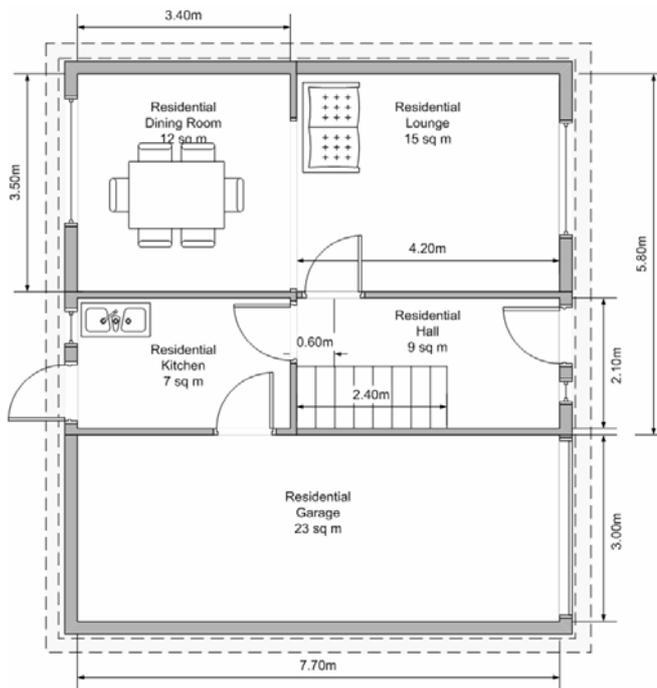
### 7.1 Geometry

In a zone fire simulation tool the basic geometry of a building is divided up into a collection of individual spaces, usually referred to as rooms or compartments. The structure of the IFC Model allows each individual space to be identified through the **IfcSpace** entity.

Using the ifcXML Parser, **IfcSpace** entities are mapped to zone fire simulation rooms with the **IfcSpace** entity label used to identify the name of the room in the fire simulation input file. The limitation of this mapping process is that spaces bounded by complex wall geometries will not convert well into a zone fire simulation tool that expects rectangular footprints. As shown in Figure 4(a), the simple2\_001204 building includes a non-rectangular room. Since both CFAST and BRANZFIRE require rectangular-shaped rooms, the ifcXML Parser translates this room into a room of dimensions 7.45m by 6.00m. This limitation is an example of where the simplifications used by a target software tool mean that the complete details given by the IFC Model are lost in the translation.



(a)



(b)

**Figure 4.** Test buildings used to verify the IFC Model translation, (a) The simple2\_001204 building available on the BLIS website imported into MS Visio Professional 2002, (b) Ground floor of the two-storey residential building drafted in MS Visio Professional 2002.

The height of a space is obtained from the height of the bounding box of the **IfcSpace** entity. Since MS Visio is a 2D drafting tool the user cannot graphically designate the height of a space. Instead the height is specified through MS Visio's space properties dialogue box.

## 7.2 Walls

A space is normally bounded by a wall, is open to a neighbouring space or open to the outside. Solid boundaries such as walls may be constructed of one or more layers each of different materials and thicknesses, for example, an outside wall might be brick but the internal partitions might be gypsum wallboard.

In the IFC Model **IfcSpace** entities are bounded by one or more **IfcSpaceBoundary** entities. These bounding entities can be **IfcWall** entities or some other appropriate entity. The ifcXML Parser identifies the **IfcWall** entities bounding a space. The thickness of the wall is obtained from the **IfcLayer** entity that is referenced by an **IfcWall** entity. An **IfcWall** entity can consist of several **IfcLayer** entities however the current version of the ifcXML Parser will only recognise the first **IfcLayer** entity for reasons discussed here.

There are several general issues associated with the translation of bounding entities from the IFC Model to BRANZFIRE and CFAST. An **IfcSpace** generally has more than one bounding wall and these are not necessarily defined as having similar construction. BRANZFIRE and CFAST both assume that a room has a single type of wall surrounding it and this leads to a number of inconsistencies when mapping between the IFC Model and the specific zone fire simulation tool.

The material property from an **IfcWall** entity is mapped to a BRANZFIRE or CFAST room wall material. The material is presently assumed to be in the 'Description' property of the **IfcWall** entity and should be one of the named properties available in the appropriate BRANZFIRE or CFAST thermal properties database. If the **IfcWall** entity description does not match a thermal database entry then the zone fire simulation may not execute as expected.

Where no wall bounding a space has a description then the equivalent BRANZFIRE and CFAST room walls will not be assigned a material. Where the walls have a mix of descriptions, the BRANZFIRE and CFAST wall material will be assigned the description of the 'last' wall that bounds the space. The 'last' bounding wall entity will depend on how MS Visio outputs its IFC file and thus cannot be necessarily predicted or assumed to be the same each time the file is saved. Finally, if some space boundary walls are given material descriptions and others not this may lead to the CFAST wall materials not correctly matching their associated rooms. This is due to way in which a CFAST input file is formatted.

MS Visio does not appear to have the facility to create multi-layer walls. Instead a single layer is used to specify the wall thickness and because of this limitation the current parser necessarily maps an **IfcWall** and its associated

**IfcLayer** thickness to the BRANZFIRE wall lining and does not map any properties to the BRANZFIRE wall substrate.

## 7.3 Openings

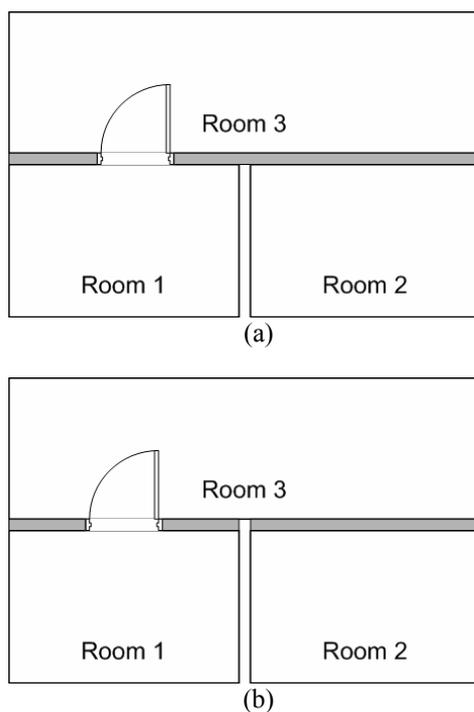
### 7.3.1 Opening types

The boundaries of a space typically contain openings that connect to neighbouring spaces or the outside. These openings might include doorways, windows, holes etc. In the case of doorways and windows, the opening may be completely filled by the door or the pane of glass. Alternatively there may be an open path through the opening if the door or window is partially or fully open.

The IFC Model specifies openings in entities using voiding entities. This allows for the identification of the parent entity for a particular opening (e.g. in which wall do we find a particular door) and also the identification of a group of openings in a specific entity (e.g. which doors form openings in a specific wall). The ifcXML Parser program identifies all these openings associated with each space during the parsing of the **IfcSpace** entities and determines the appropriate connections. The user can specify whether they want all windows and/or doors to be fully open or completely closed however it is currently not possible to specify the states of individual openings.

### 7.3.2 Connecting spaces via openings

The simple2\_001204 building allowed connections between spaces to be made by using the fact that an opening is referenced to by both parent spaces. However when a new building plan is created in MS Visio 2002 an opening only belongs to a single space making it impossible to identify the connecting space using an algorithm designed for files in the form of the simple2\_001204 building. Instead a second algorithm was developed that uses the fact that we can identify the space boundary between two spaces and find if a wall belongs to that boundary. If a wall does exist on the boundary, we can check if any other entities such as doors and windows void the wall.



**Figure 5.** IFC Model translation of walls that bound common spaces, (a) single common wall, (b) individual walls, wall between Room 1 and Room 2 not shown.

assumed by the ifcXML Parser that a single slab defines the properties for the whole floor which means that it is not possible to have different properties for different rooms. The lack of ceiling and floor entities in the IFC Model presents difficulties where we might want to consider these independently from the slab. For example we might be interested in knowing the flammability characteristics of a floor system which might not be the same system in different spaces.

This second algorithm has particular issues regarding bounding walls that are common to several spaces. If we consider the geometry shown in Figure 5(a) where the wall is common to Room 1 and Room 2 the ifcXML Parser would determine that Room 3 is next to both Room 1 and Room 2 via their common wall. When finding which two rooms the door connects, the ifcXML Parser would be unable to determine whether the door links Room 3 with Room 1 or Room 3 with Room 2 simply from knowing only that the rooms are next to each other. Currently the way to overcome this problem is to have separate walls bounding the smaller compartments such as shown in Figure 5(b) in which case the door can only connect Room 3 with Room 1 via the wall that only bounds these two rooms. An alternative solution would be to consider the relative coordinates of the door with respect to the connecting rooms in order to determine the appropriate link and this may be implemented in a future version of the ifcXML Parser.

What is not clear regarding this discrepancy between the simple2\_001204 building and the house draughted in MS Visio is whether the implementation in MS Visio is intentionally different to the BLIS example. The current version of the ifcXML Parser assumes that there are no common walls between spaces so that floor plans appropriately drafted in MS Visio are correctly translated.

## 7.4 Floors and ceilings

In the IFC Model **IfcSpace** entities do not have an explicit associated ceiling or floor entity. Instead the floor and ceiling properties for each room are obtained from **IfcSlab** entities. It is

To define the floor and ceiling slabs, the user creates an MS Visio 2002 'slab' from the 'Wall, shell and structure' stencil. The slab can be located anywhere on the plan and does not necessarily need to cover the whole floor plan. The user then needs to specify the 'Slab type' to indicate whether the slab is a 'Floor' or 'Roofslab'. The user can also specify the thickness of the slab and the material.

## 7.5 Fire

In order to conduct a simulation, the software needs to be able to determine the characteristics of the fire itself. Fundamentally this could be an ignition source and the thermo-physical properties of the fuels however many fire simulation tools are unable to model the ignition and fire spread mechanisms particularly for mixed fuels in complex arrangements. As an alternative many zone fire simulation tools typically require the user to specify heat release rate data to define the fire. In many situations a fire in a building is the result of the burning of the contents and these contents are often items of furniture. The IFC Model includes a variety of furniture items and therefore these can be used as suitable fire sources.

It is not possible to directly specify the heat release rate of the contents of a building using MS Visio and this is likely to be the case with any other commercial CAD tool. Instead heat release rate data needs to be added from an external source and an IFC Model PSD heat release rate data format has already been defined elsewhere [9]. The heat release rate data can be from any convenient source of which one such source is the FireBaseXML database [15]. A software tool called SelectFire has been developed to interrogate such a database is available and a translation has been written to convert a FireBaseXML record into a PSD.

Items of furniture without heat release rate data are parsed but not added to the BRANZFIRE or CFAST input files. With the translation to CFAST, if the IFC file includes more than one item of furniture with an attached heat release rate data then the 'last' entity found will be used. However with BRANZFIRE, if the IFC file includes more than one item of furniture with an attached heat release rate data then the ifcXML Parser adds each entity to the room. The 'first' furniture entity found will specify the location of the fire but the user can easily modify this in BRANZFIRE prior to running a simulation. In each case the 'first' and 'last' furniture entities will depend on how MS Visio outputs the IFC file and thus cannot be necessarily predicted or assumed to be the same each time the file is saved.

BRANZFIRE expects the energy output from the fire to be expressed in kW and the time in seconds as supplied by FireBaseXML. CFAST expects the energy output from the fire to be expressed in Watts so the PSD must be appropriately interpreted during translation. Also CFAST only allows 20 data points in its heat release rate curve and if more points are specified then only the first 20 will be used.

Where appropriate, the fire is positioned in the room using the location provided in the IFC file. The dimensions of the fire are taken from the bounding box of the `IfcFurniture` entity. For complex shapes this may only be an approximate representation. The user would need to edit the simulation input if they wanted the dimensions of the fire to change over time, for example, if the area of the fire changed with time. This is one example of where the IFC Model is currently unable to deal with dynamic views of a building.

## 8. Future work

The complexity and scope of the IFC Model means that there are still considerable enhancements that can be added to the ifcXML Parser. For example, at the time of writing the translation to the zone fire simulation software does not fully process `IfcStairs` entities. It should also be possible to allow the user to specify the opening state of individual windows or doors through MS Visio. There is also the need to test additional buildings such as multi-storey structures and buildings that have been draughted in other IFC Model compatible software where the files are either in STEP or ifcXML format.

The current IFC Model has entities that can be used to specify components of smoke control systems such as fans and dampers however there may be a need to enhance these entities so as to encompass all of the available components and to describe all relevant properties. In the longer term the core IFC Model could be extended to include specific fire safety systems such as detection components, suppression systems etc. Some of this could be achieved using the PSD capabilities included in the IFC Model.

Future interfaces to other fire simulation software are possible and these might include CFD fire simulation software or people movement simulation tools. It might also be desirable to make the translation of an IFC Model file integral with the fire simulation software rather relying on a separate parser tool. This would need access to the

fire simulation software's source program and may require considerable effort in order to integrate the parsing processes and a suitable user interface within the fire simulation software.

## 9. Conclusions

The work described in this paper demonstrates how the IFC Model is a general way in which to describe buildings and the means of using this model to enable the sharing of the description to fire zone simulation software. There is still much work to do including the expansion of the IFC Model to include more fire-specific information either in the core model or by using PSDs, enhancing the interpretation of the IFC files for fire zone simulation software and widening the scope of models that can use the IFC descriptions.

## References

1. Mowrer F W, Williamson R B. *Room fire modeling within a computer-aided design framework*. International Association for Fire Safety Science. 2nd International Symposium. June 13-17, Tokyo, Japan, pp.453-462, 1988.
2. Quintiere J G. *Compartment fire modeling*. Chapter 3-5, The SFPE handbook of fire protection engineering (3<sup>rd</sup> ed.), p. 3-162 – 3-170, 2002.
3. Olenick S M, and Carpenter D J. *An updated international survey of computer models for fire and smoke*. Journal of Fire Protection Engineering, 15 (2) pp.87-110., (see <http://www.firemodelsurvey.com>)
4. Spearpoint M J. *The potential impact of building product models on fire protection engineering*. Fire Protection Engineering (in press).
5. Ito K. *General product model and domain specific product model in the A/E/C industry*. Proc. 2<sup>nd</sup> Congress on Computing in Civil Engineering, Atlanta, GA. Vol. 1 p. 13-16, 1995.
6. Liebich T, Wix J, ed. *IFC technical guide, Industry Foundation Classes – Release 2x*, International Alliance for Interoperability, 2000.
7. ISO 10303-21:2002 *Industrial automation systems and integration - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure*.
8. Liebich T. *XML schema language binding of EXPRESS for ifcXML*. MSG-01-001(Rev 4), International Alliance for Interoperability, 2001.
9. Spearpoint M. *Properties for fire engineering design in New Zealand and the IFC building product model*. Proc. 20<sup>th</sup> International Conference on Information Technology in Construction, CIB Publication 284, ISBN 0-908689-71-3, 2003.
10. BLIS Project Companies. *Building Lifecycle Interoperable Software - Project Brief*. <http://www.blis-project.org> (Accessed March 2003).
11. Bukowski R, Peacock R, Jones W, Forney C. *Technical Reference Guide for HAZARD I*. NIST Handbook 146, Volume II. Gaithersburg, National Institute of Standards and Technology, 1989.
12. Bukowski R. *Applications of FASTLite*. Proc. Computer Applications in Fire Protection Engineering, Technical Symposium, June 20-21, 1996, Worcester, MA.
13. Wade C A. *BRANZFIRE Technical reference guide*. Study Report No. 92, ISSN: 0113-367, BRANZ, Porirua City, 2002.
14. Spearpoint M J and Smithies J N. *Practical comparison of smoke detector sensitivity standards*. Proc. 11<sup>th</sup> International Conference on Fire Detection, AUBE '99 conference, Duisburg, Germany. 1999.
15. Spearpoint M. *The development of a web-based database of rate of heat release measurements using a mark-up language*. Proc. 5<sup>th</sup> Asia-Oceania Symposium on Fire & Technology, Newcastle, Australia. 2001.