Assigning doors & windows to spaces using IFC R2.0

DRAFT 3 – 17.12.1999

NOTE: In this Draft 2 I have stripped away the analysis of the different possibilities to solve the question and concentrated on IfcRelContains. I hope Draft 2 will be the basis for an implementer's agreement on this topic. For the complete analysis, please refer to Draft 1, dated 19.11.1999 (File name: IFCR2_AssigningDoorsAndWindowsToSpaces_991119_jh.pdf)

Table of contents:

Why do we need this information?

In the IFC R2.0 specifications there is no designated method for assigning doors and windows to a specific space. This functionality is required in many types of applications such as thermal analysis, escape route planning, security zone analysis and quantity takeoff

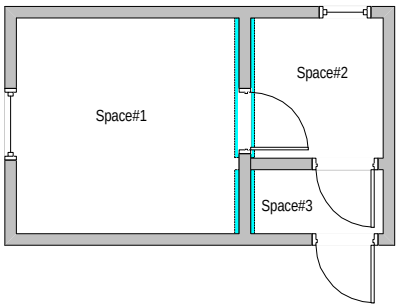
Thermal analysis

In the IFC model spaces are wrapped with space boundaries (IfcSpaceBoundary). The main challenge for thermal applications is to find out what can be found on the other side of the space boundary. Is it another space or outside air, and if it is another space then which space is it?

There can be one or more spaces behind a single space boundary and the program has to calculate what part of the space boundary is against what space. This analysis is so specialized that we cannot expect the IFCs to provide this information; the design applications would simply not know how to do this.

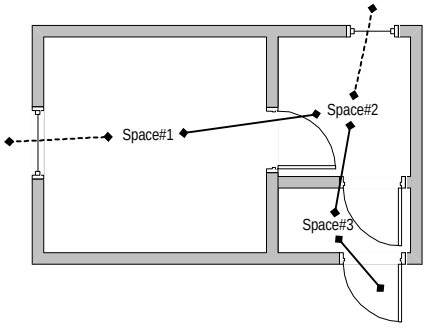
After having divided the space boundary into areas that correspond to what can be found behind the walls the next challenge is to assign doors and windows to those areas. At the moment this is unnecessary difficult, because there is no connection between the doors/windows and the spaces/space boundaries. It would be relatively easy for design applications to provide this information and thus make the task for thermal analysis programs much easier.

Author: Jiri Hietanen (jiri.Hietanen@qpartner.com)



Space flow analysis

The needs for space flow and zone analysis are less complicated than for thermal analysis. The main task is to find out how spaces are connected to each other. At the moment this would require the application to analyse the IFC geometry, and analysing 3D geometry is not what these applications normally do.



Quantity takeoff

Also quantity takeoff has its limited benefits from assigning doors and windows to spaces. Although the IFCs currently don't provide detailed information about the quantities of the space surfaces it is useful to know the dimensions of opening elements making 'holes' into the space surfaces.

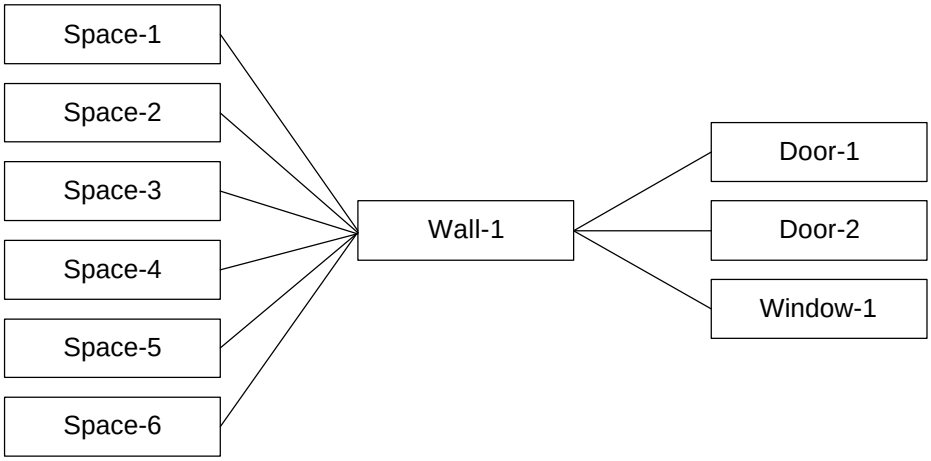
Knowing into which space doors and windows are installed is also valuable information for scheduling.

Why do we need a special relationship for this information?

In this paragraph I try to elaborate why a direct relationship between spaces and doors / windows is a better approach than deriving the same information from the model.

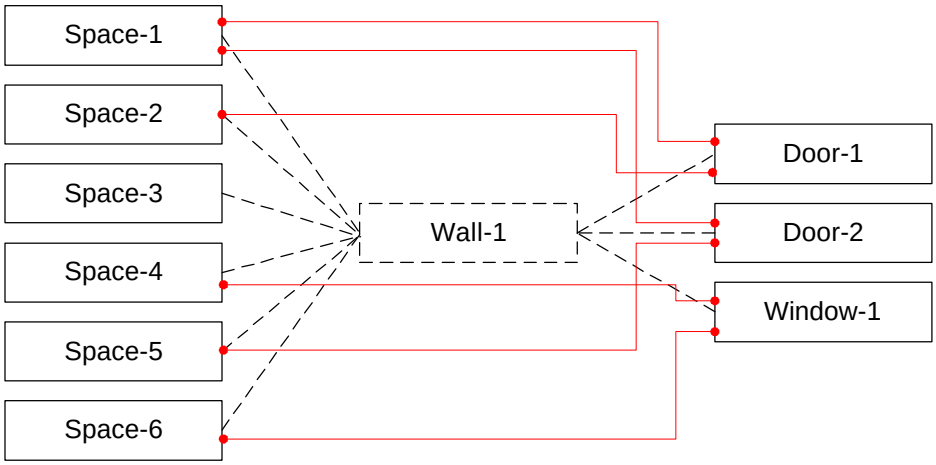
Let us first think about the existing relationships between spaces and wall, and the relationships between doors/windows and walls.

Author: Jiri Hietanen (jiri.Hietanen@qpartner.com)



This is the kind of relationships we get from examining the IfcSpace – IfcSpaceBoundary – IfcRelSeparatesSpaces – IfcWall relationship, and the placement of doors and windows through IfcDoor – IfcLocalPlacement – IfcOpeningElement – IfcLocalPlacement – IfcWall.

But how can we associate the doors and windows with the spaces, when all we know is that they are all connected to the same wall? What are the two spaces that e.g. Door-1 connects, are they Space-4 and Space-3 or Space-1 and Space-2 or …? Currently the only way to find this out is to examine the geometry of the spaces relative to the geometry of the doors and windows. This can be done, and Granlund and Secom have already done it, but it demands quite a lot from the receiving application



Adding direct relationships between the spaces and the doors/windows would significantly lower the bar for taking advantage of this information. In the strictest sense it is redundant information, but in my opinion the benefits are bigger than the trouble.

Proposed solution: IfcRelContains

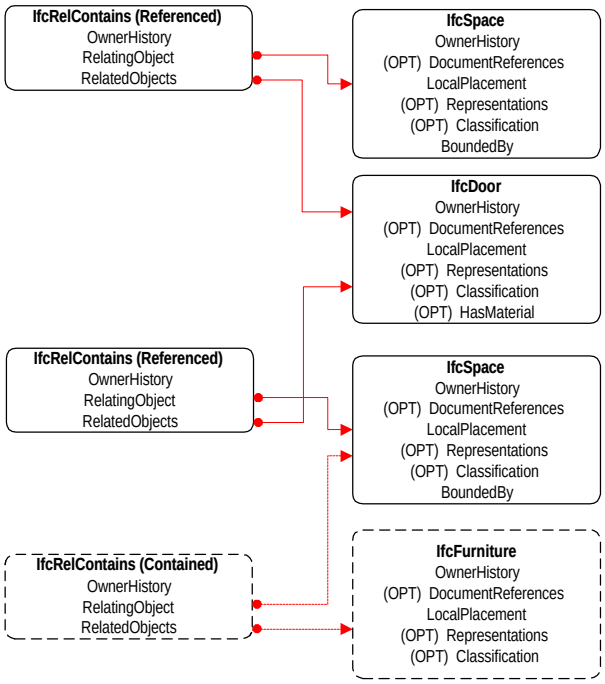
The IfcRelContains objectified relationship allows us to form any arbitrary group of objects that is contained by another object. One clear example of this would be that furniture is contained by a space. It is in the nature of containment that one piece of furniture can only be contained by one space.

Author: Jiri Hietanen (jiri.Hietanen@qpartner.com)

However, there is an attribute `IfcRelContains.ContainedOrReferenced` that allows us to use IfcRelContains also for referencing. This means that for each space we could make a group that contains the doors and windows referenced by that space. One door could be referenced by two different spaces and this would tell us that this door is between the two spaces.

The IfcRelContains that references the doors and windows of a specific space would be identified by these three attribute values:

1. `RelatingObject` = reference to the IfcSpace instance in question
2. `RelationshipType` = "`SpaceContainer`"
3. `ContainedOrReferenced` = "`Referenced`"



I have earlier tried to propose a solution where one space could have several different groups that are related to it. This way the doors and windows could be in one group and e.g. HVAC systems in another group. Since it is not clear that this functionality would really be used in IFC R2.0 implementations I propose that we make our lives easier and forget about that idea for now. One space would have exactly one group (IfcRelContains) that contains everything that is referenced by that space. There would also be exactly one group (IfcRelContains) that contains everything that is contained by the space, e.g. furniture.

Programming: IfcRelContains

Thermal analysis

1. Analyse the geometry to find out what can be found behind one space boundary of the space. The result is a list of boundaries with an area and information what can be found on both sides of that boundary, e.g. a space or outside air. These boundaries are sub areas of the original space

Author: Jiri Hietanen (jiri.Hietanen@qpartner.com)

boundary.
The next task is to find the openings, doors and windows that are inside one of these boundaries.

2. Use the INVERSE relationship IfcSpace.Contains in the first space to locate the IfcRelContains object that holds references to the doors and windows for that space.
3. Get the IfcRelContains.RelatedObjects –list that contains the doors and windows of the first space. Filter out everything else than doors and windows.
4. Get the IfcRelContains object that holds references to the doors and windows of a space behind the boundary
5. Get the IfcRelContains.RelatedObjects –list that contains the doors and windows of the second space. Filter out everything else than doors and windows.
6. Compare the two lists. The doors and windows that can be found in both lists are between the two spaces
7. After having gone through all space boundaries of the first space there will be a list of doors and windows that have not been used yet. The doors and windows that can only be found in the list of the first space are between the first space and outside air.
8. (To get the right are for the boundary the area of the opening elements has to be deducted from the boundary area. This requires that the program looks up the IfcOpeningElement for the door or window and looks up or calculates the area of the opening)

Space flow analysis

1. Use the INVERSE relationship IfcSpace.Contains in the first space to locate the IfcRelContains object that holds references to the doors and windows for that space.
2. Get the list IfcRelContains.RelatedObjects that contains the doors and windows of the first space. Filter out everything else than doors and windows.
3. Iterate through the IfcRelContains objects (of the right type) in the model and get the IfcRelContains.RelatedObjects list for each of them
4. If the same door or window can be found in both lists the IfcRelContains. RelatingObject points to the second space. This means that the door or window is between the two spaces. If the door or window can be found in none of the other lists this means that the door or window is between the first space and outside air

Quantity takeoff

1. Use the INVERSE relationship IfcSpace.Contains in the space to locate the IfcRelContains object that holds references to the doors and windows for that space.
2. Get the list IfcRelContains.RelatedObjects that contains the doors and windows of the space. Filter out everything else than doors and windows.

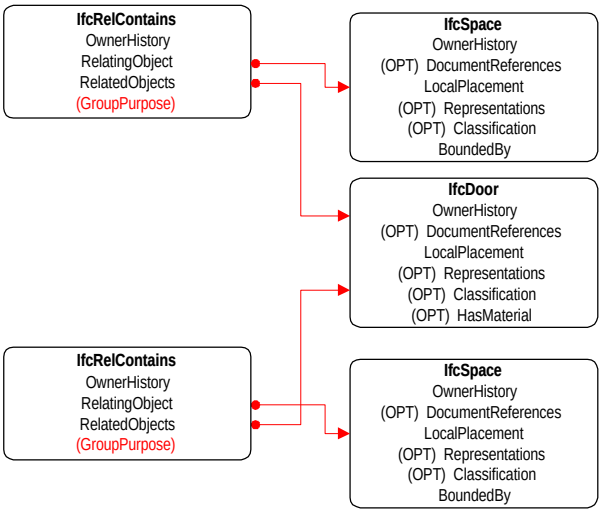
Author: Jiri Hietanen (jiri.Hietanen@qpartner.com)

Future IFC releases

For future IFC releases having the ability to assign more than one logical group to a space should be considered. If there is only one group per space the current system is sufficient. The following proposals give different alternatives to how this could be solved.

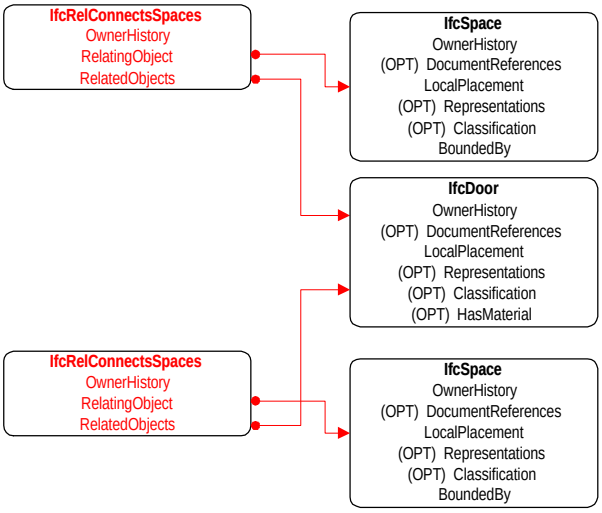
Add purpose for IfcRelContains

One possibility is to add an enumeration / string to IfcRelContains that identifies the different uses for IfcRelContains to make absolutely clear what the IfcRelContains – group is used for. The inverse attribute `Contains` on the IfcObject level is currently `SET [0:2] OF IfcRelContains`, it would have to be relaxed to `SET [0:?] OF IfcRelContains.`



Subtype IfcRelContains

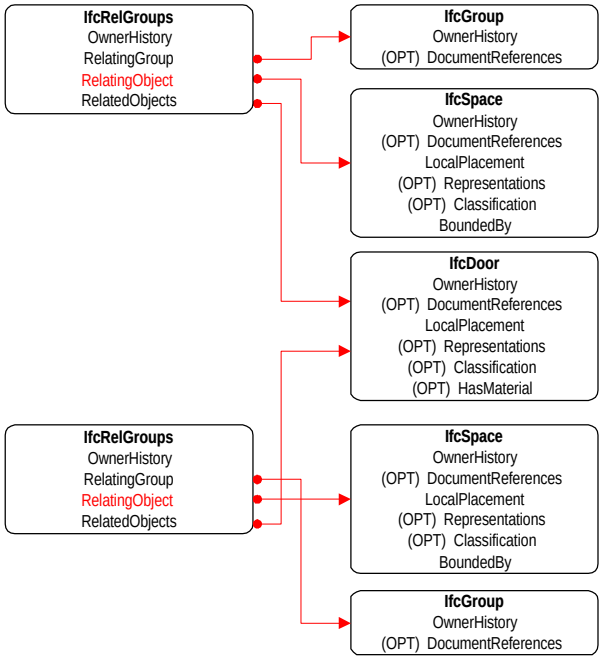
One possibility is to make a special subtype of IfcRelContains that takes care of the space to door / window relationship. . The inverse attribute `Contains` on the IfcObject level is currently `SET [0:2] OF IfcRelContains`, it would have to be relaxed to `SET [0:?] OF IfcRelContains.`



Author: Jiri Hietanen (jiri.Hietanen@qpartner.com)

Add optional RelatingObject to IfcRelGroups

One possibility is to use IfcRelGroups instead of IfcRelContains for this purpose. This would be semantically clearer, but the necessary RelatingObject attribute is currently missing from IfcRelGroups. If RelatingObject is added it should be optional.
IfcGroup has a GroupPurpose that could be used to identify the group.



Author: Jiri Hietanen (jiri.Hietanen@qpartner.com)