buildingSMART
International Alliance for Interoperability

IFC Implementation Agreement Space Boundary

Overview on the common agreements for implementing space boundaries

April 2009 / proposed amendment March 2010

Initial summary: Karl-Heinz Häfele, edited: Thomas Liebich

Space Boundary Implementation Agreements

Consolidated results from various initiatives and meetings on implementing space boundaries

Initiatives:
buildingSMART Implementer Support Group
buildingSMART German Speaking HVAC group
OGC/buildingSMART Alliance AECOO-1 testbed
ERDC ENERGie project
European Integrated Project InPro

Meetings (selection):
09.03.2009 / FZ Karlsruhe, Olof Granlund
ISG meeting 11.03.2009, including
LBNL, Graphisoft, DDS, Granlund, FZK, AEC3,
Teleconference 12.03.2009, DigitalAlchemy, CIFE/GSA

Distributed to various groups for comments until April 1st

SB implementation agreements – amendment #1

Proposed amendment #1 – March 2010

- There are cases where the restriction of not using inner loops leads to results where multiple split space boundaries would have to be created (e.g. around a column touching a ceiling, etc.)
- The Helsinki agreements of April 2010 had been misinterpreted that no inner loops are allowed (correct: they are not allowed for openings with/without doors and windows)
- This leads to the amending clarifications and additions, see pages 15, 16, 17.

Amendment March 2010

General thoughts

Space Boundaries are needed to support different tasks, e.g.

- **Energy Calculation,**
- **Lighting Calculation**
- **Indoor Navigation**
- **Quantity Take Off and**
- **Facility Management**

Different tasks require different kinds of space boundaries

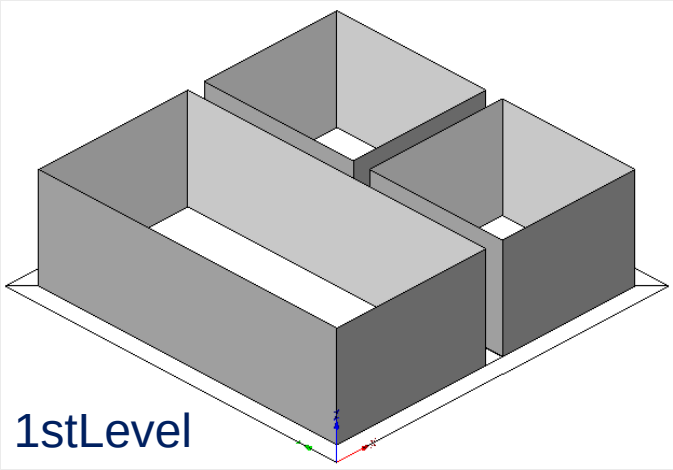
- **All should be derived from the same principles**
- **More granular kinds are derived from more general kinds**
- **A “kind” of space boundary is also referred to as “level”**

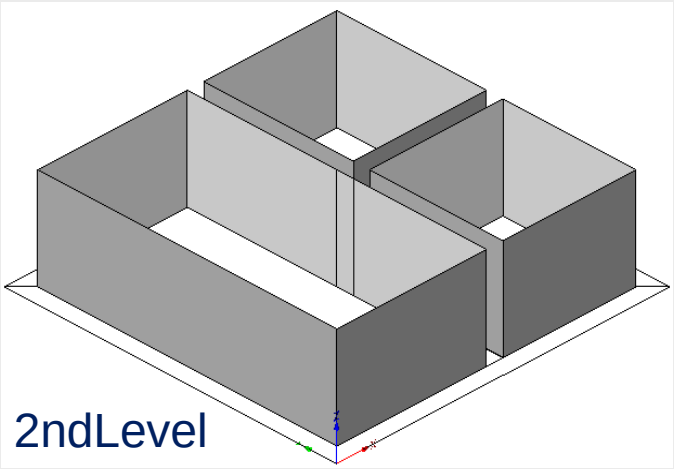
Principles for space boundaries

Space Boundaries have to be defined as simple, clearly and redundant free as possible

- **Higher level space boundaries are "specializations", not "contradictions" of lower level space boundaries**
- **There should be only two levels of space boundary implementations**
  - **Space surfaces boundaries, also referred to by 1st level**
  - **Thermal space boundaries, also referred to by 2nd level**
- **Each IFC exchange file can ONLY contain boundaries of ONE single level, either 1st or 2nd.**
  - **Note: the term 2nd level consumes all special cases needed for thermal analysis (it combines and hides 2nd, and 3rd level)**

1stLevel & 2ndLevel Space Boundaries (SB)





Differences between SB levels

- Influenced by “what is on the other side”
  - 1stLevel - no influence,
  - 2ndLevel - influence

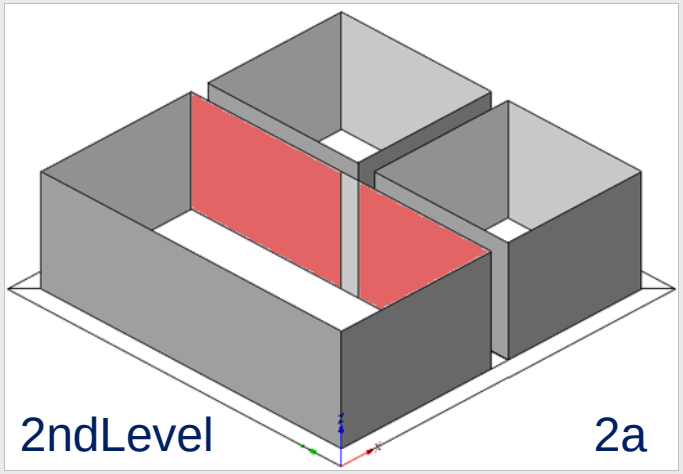
- Reflected in the IFC Header

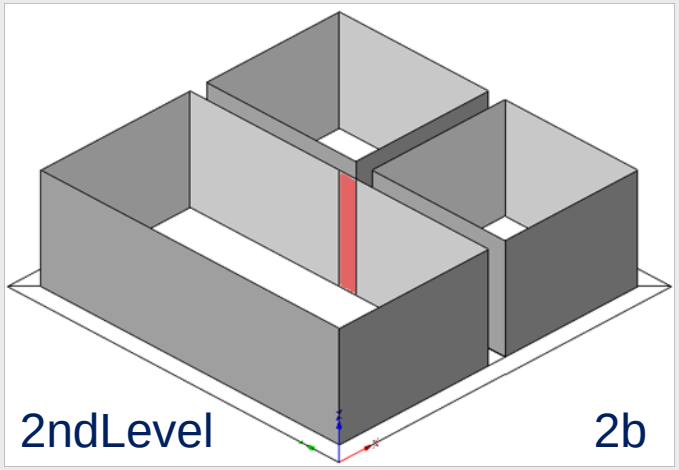
  SpaceBoundary1stLevelAddOnView

  SpaceBoundary2ndLevelAddOnView

  No combination allowed

Sublevels of 2nd level Space Boundaries





Differentiation within 2nd Level

- Influenced by “what kind of element is on the other side”
  - 2a) there is a space behind
  - 2b) there is no space behind, but a physical element

- Note: 2b is also called 3rd level SB

Reflection within the IFC exchange file -1-

IFC Header

- File containing only 1stLevel space boundaries

```
HEADER;
FILE_DESCRIPTION(('ViewDefinition [CoordinationView,
   SpaceBoundary1stLevelAddOnView]'),'2;1');
```

- File containing only 2ndLevel space boundaries

```
HEADER;
FILE_DESCRIPTION(('ViewDefinition [CoordinationView,
   SpaceBoundary2ndLevelAddOnView]'),'2;1');
```

Reflection within the IFC exchange file -2-

IFC Space Boundary Objects within IFC File

- 1stLevel
  - *IfcRelSpaceBoundary.Name* = “1stLevel”
  - *IfcRelSpaceBoundary.Description* = $ (i.e. NIL)

```
#5= IFCRELSPACEBOUNDARY('2gOpAsSZf0Zv7F6pkKDuGM',#1,'1stLevel',$,
#1100,#2100,#15,.PHYSICAL.,.INTERNAL.);
```

- 2ndLevel
  - *IfcRelSpaceBoundary.Name* = “2ndLevel”
  - *IfcRelSpaceBoundary.Description* = “2a”, or “2b”

```
#6= IFCRELSPACEBOUNDARY('2gOpAsSZf0Zv7F6pkKDuGN',#1,'2ndLevel',
'2a',#1100,#2100,#16,.PHYSICAL.,.INTERNAL.);
#7= IFCRELSPACEBOUNDARY('2gOpAsSZf0Zv7F6pkKDuGO',#1,'2ndLevel',
'2b',#1100,#2100,#17,.PHYSICAL.,.INTERNAL.);
```

Building Elements having Space Boundaries

Elements in an IFC file that have to have space boundaries

- Walls (incl. Curtain Walls)
- Slabs
- Roofs
- Columns
- Windows and Doors
- Openings (Virtual Elements)
- Space Separators (Virtual Elements)

Elements that do not have space boundaries

- Beams
- Stairs and Ramps (external)
- Building Element Proxies

Container Elements providing space boundaries

Container elements are elements with parts, such as

- *IfcWall* (when decomposed into *IfcBuildingElementPart*'s)
- *IfcRoof*
- *IfcCurtainWall*

The container itself has the boundary geometry

- the *IfcCurtainWall* (or other container) has the *IfcRelSpaceBoundary* attached
- even if the *IfcCurtainWall* has own elements as parts, and those parts have geometry only the *IfcCurtainWall* has boundaries
  - The space boundary of the *IfcCurtainWall* is potential simplified
  - Having a space boundary per every lintel, etc. would be an overkill

Connection geometry for 1st and 2nd Level SB

Limited to:

- Only connection geometry at the *RelatedSpace*
- Only *IfcConnectionSurfaceGeometry* (no Point, no Line)
- Same for 1st and 2nd level SB

```
#1851= IFCRELSPACEBOUNDARY('0eY2BBnPLEhxpxN_7YB5l1',#13,
'1stLevel',$,#762,#362,#1850,.PHYSICAL.,.EXTERNAL.);

#1850= IFCCONNECTIONSURFACEGEOMETRY(#1846,$);
```

Geometric items for 1st and 2nd Level SB

Geometric items of the connection surface geometry

- In case of 1st Level
  - *IfcSurfaceOfLinearExtrusion* with trimmed curves (line or arc)
  - *IfcCurveBoundedPlane* with no restrictions of outer bound
  - *IfcFaceBasedSurfaceModel* with no restrictions
- In case of 2nd Level
  - *IfcCurveBoundedPlane* with restrictions
    (only polyline as outer boundary)
  - *IfcFaceBasedSurfaceModel* with no restrictions
    Curved space boundaries are faceted, a recommendation of number of facets is made (36 per 360', or 1 of 10')

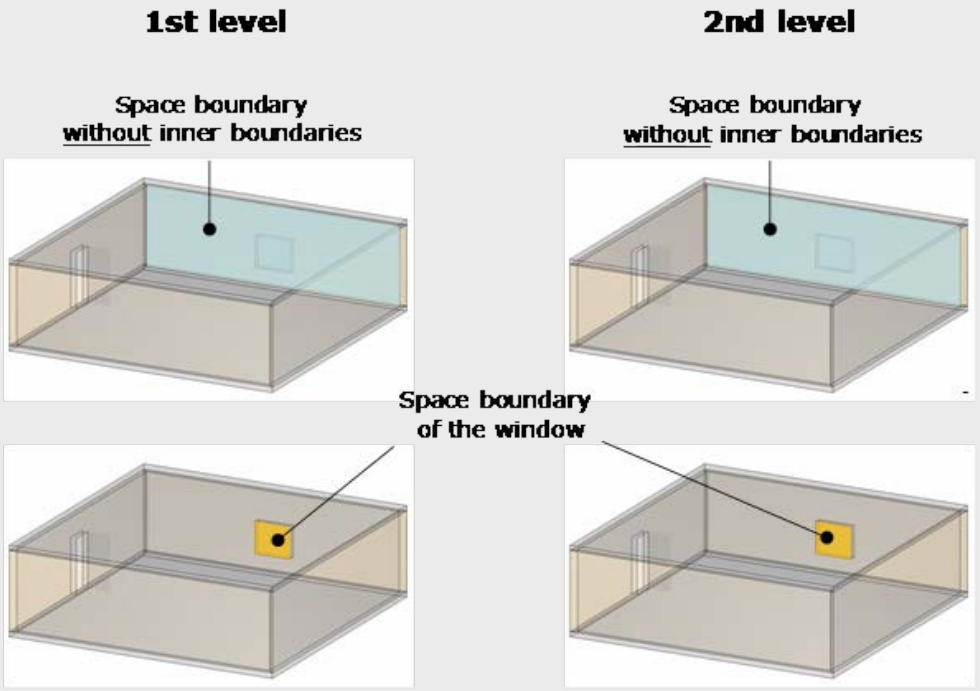
Openings (including door/window) having SB's

Openings (including door/window) have space boundaries

- They do not generate "holes" or "inner loops" in the space boundaries of the walls or slabs in which they are contained
- Same for 1st and 2nd level space boundaries

Solution is the same for an *IfcOpening* (without window/door) and an *IfcOpening* with window and door as fillings.

In case of *IfcWindow* and *IfcDoor* – the space boundary is then attached to the window and door, and NOT to the opening.



In case of *IfcOpeningElement* without a filling (door or window) – the space boundary is attached to the opening

SB for holes NOT created by openings

Space boundaries of building elements that have holes NOT generated by an opening (i.e. no *HasOpenings* inverse relationship) as defined in previous page:

Holes CAN be represented by EITHER

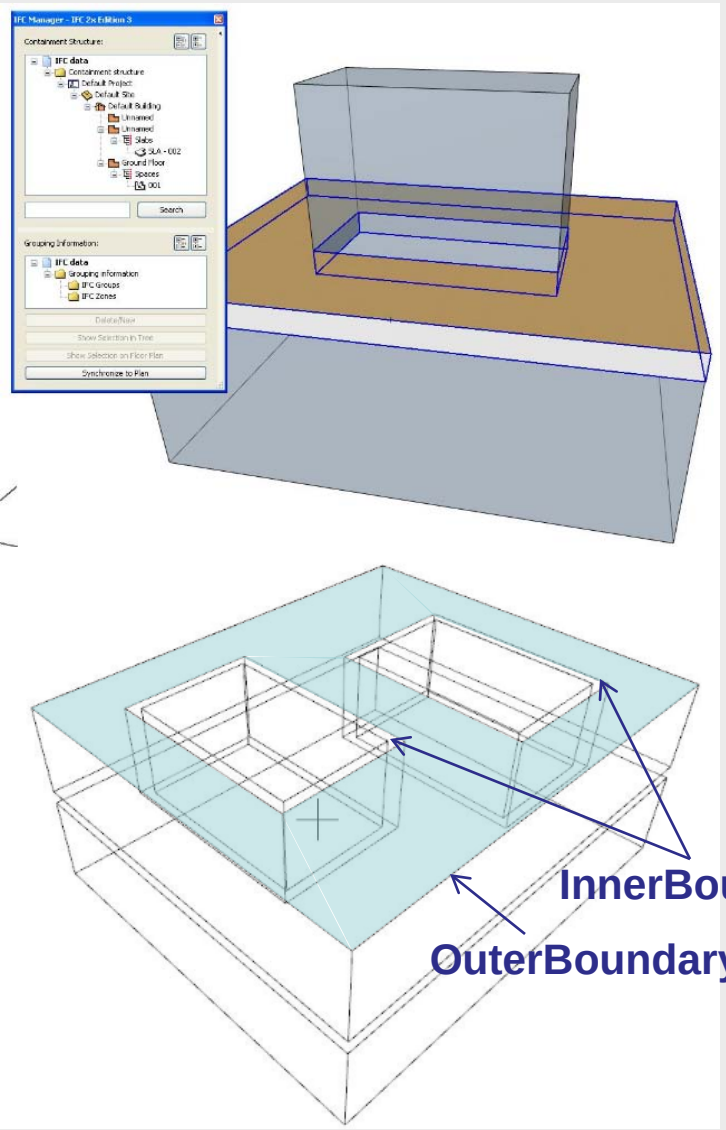
- **"inner loops" – SIZEOF (*IfcCurveBoundedPlane.InnerBoundaries*) > 0**
- **splitting the space boundary into smaller SB's around the hole**

Same applies to 1st and 2nd level SB

(see next pages for examples)

Amendment March 2010

SB for holes (NOT openings) using inner loops



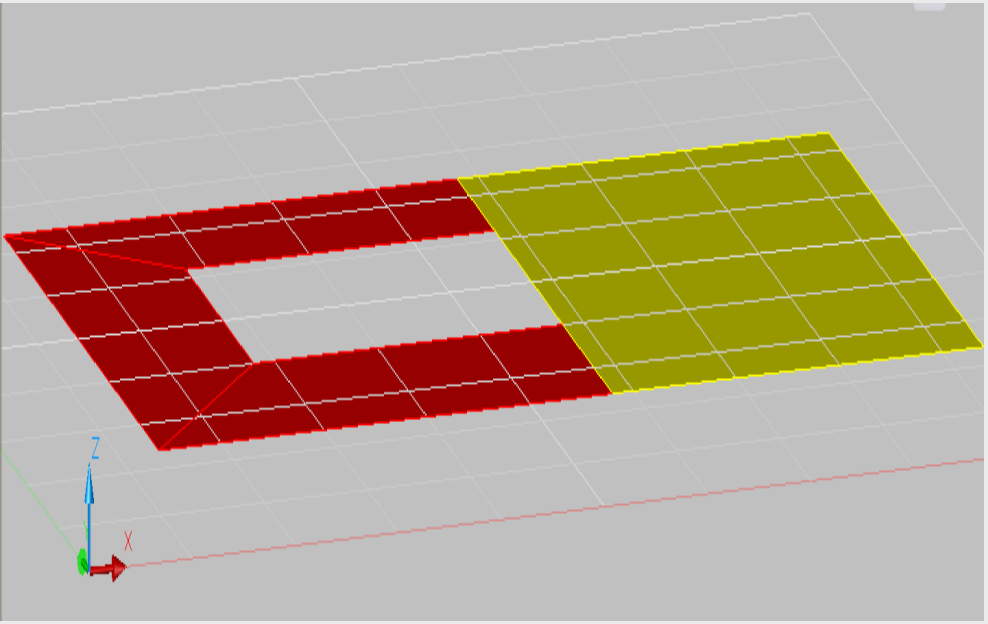
Example: multi-storey space through a slab (no opening)

Example: a big space with fully enclosed inner spaces

NOTE: topological constraints for inner loops (not overlapping each other and not overlapping outer boundary) have to be preserved

Amendment March 2010

SB for holes (NOT openings) by splitting SB's



Example: slab with hole (not opening), exported by multiple space boundaries (all without "inner loop").

NOTE Number of space boundaries depends on splitting algorithm.

SUMMARY of Amendment March 2010:

1.) agreement not to subtract opening space boundaries from "host" surface remains unchanged
2.) clarify that other cases can be exported by using "inner loops" or by splitting the space boundaries

Amendment March 2010

Virtual Space Boundary – Virtual Element

Detailed implementation questions

- How to implement *IfcRelSpaceBoundary.PhysicalOrVirtualBoundary*
  - Current Implementation Guide: If the attribute *PhysicalOrVirtualBoundary* is set to VIRTUAL the bounding element is an *IfcVirtualElement*, or an *IfcOpeningElement*
  - Note: this ignores the *IfcRelSpaceBoundary.WR* (would generate an error (is already fixed for the next release of IFC)

- Add to it:
  - → If there is no related element, the attribute *PhysicalOrVirtualBoundary* should be NOTDEFINED

Internal or external (1. & 2. Level)

Detailed implementation questions

- How to implement *IfcRelSpaceBoundary.InternalOrExternalBoundary*
- Should be recalculated without considering the Common Properties (IsExternal) of the related Building Element
  - Valid setting of the *InternalOrExternal* flag : EXTERNAL, INTERNAL, NOTDEFINED for 1st level, and EXTERNAL, INTERNAL for 2nd level
  - In the case of 1stLevel Space Boundaries, if the boundary is internal and external the attribute should be set NOTDEFINED
  - In the case of 2ndLevel Space Boundaries: it has to be either EXTERNAL, or INTERNAL

Completeness (1stLevel, 2ndLevel)

Space boundaries shall completely bound any space

- Always air tight
- Always complete, sum of (PHYSICAL, VIRTUAL, NOTDEFINED) fully encloses a space
- Boundaries of doors, windows, openings overlap the wall, slab SB's

Correct surface orientation

- Surface normals always point outward of the space (into the material)