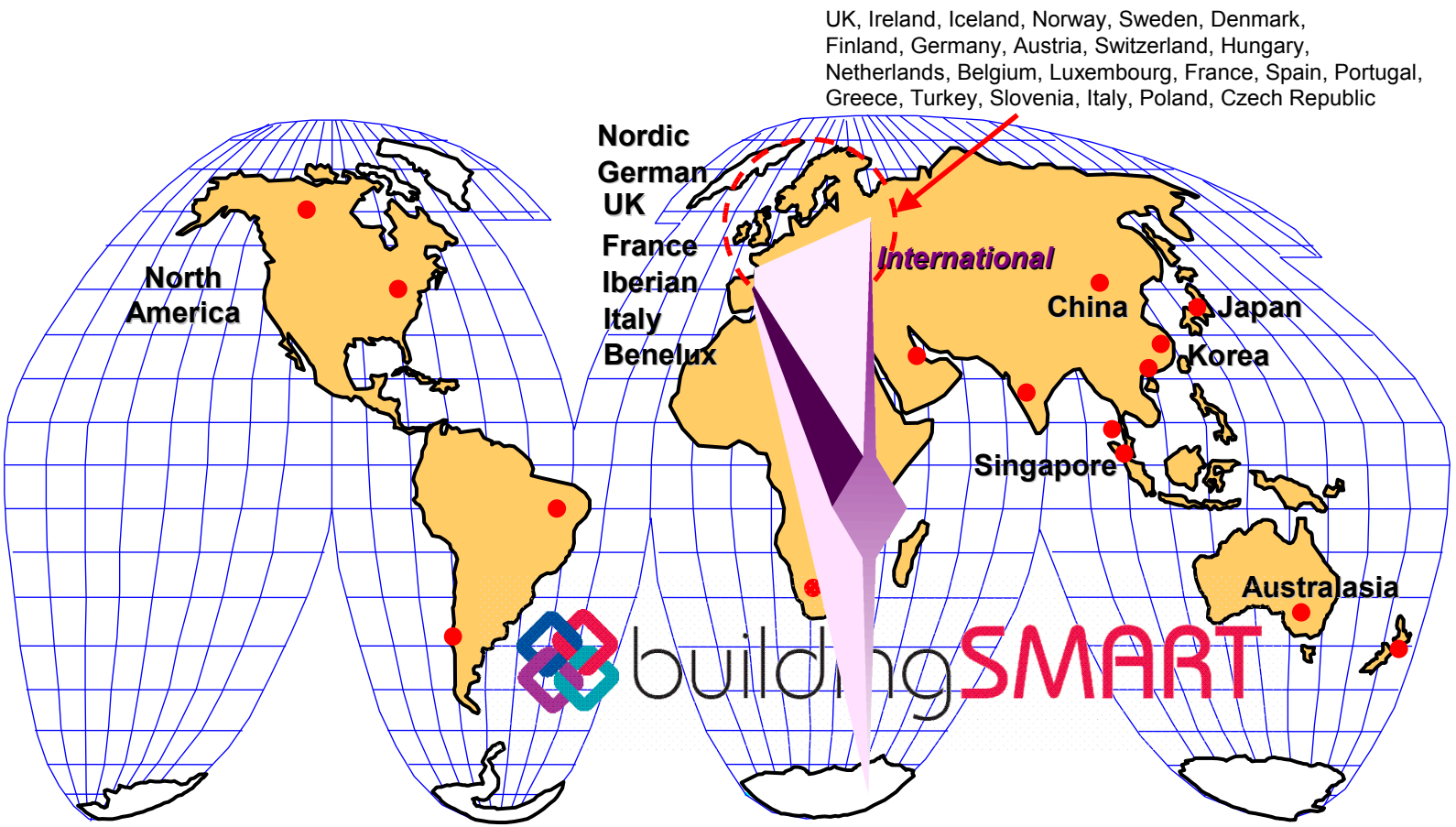

What is IFC?
Jeffrey Wix

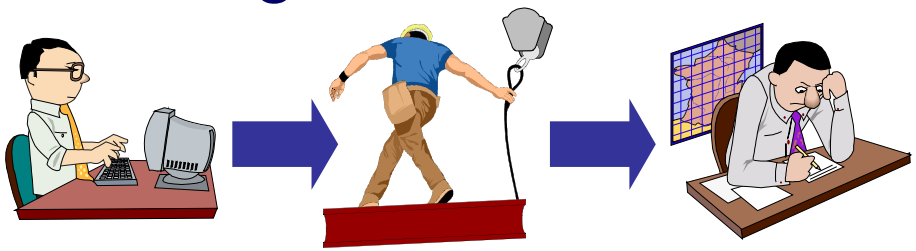
# What is buildingSMART?



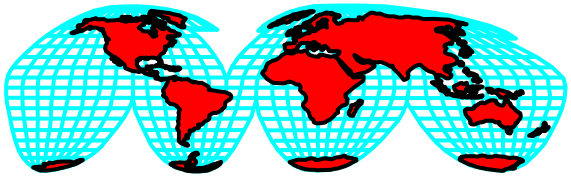

| 13 Chapters | ~35 Countries | >550 Organisations |
|---|---|---|

# Mission

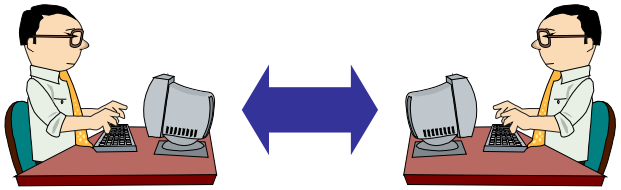
- To define, promote and publish a specification for sharing data:



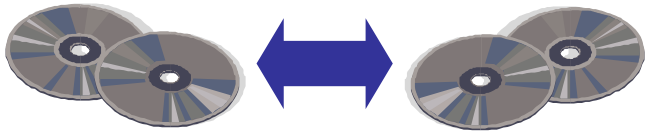
- throughout the project life-cycle



- globally



- across disciplines



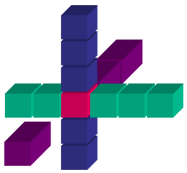
- between software applications

# Objectives

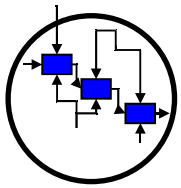
- Set the standard for object-based data exchange and sharing of virtual buildings:

  - IFC schema
    - **the comprehensive foundation specification for information**

  - Dictionary
    - **consistent names and definitions for different properties to support catalogs, classification systems etc.**



  - Exchange Requirements
    - **units of the IFC model tailored for use in particular business situations**

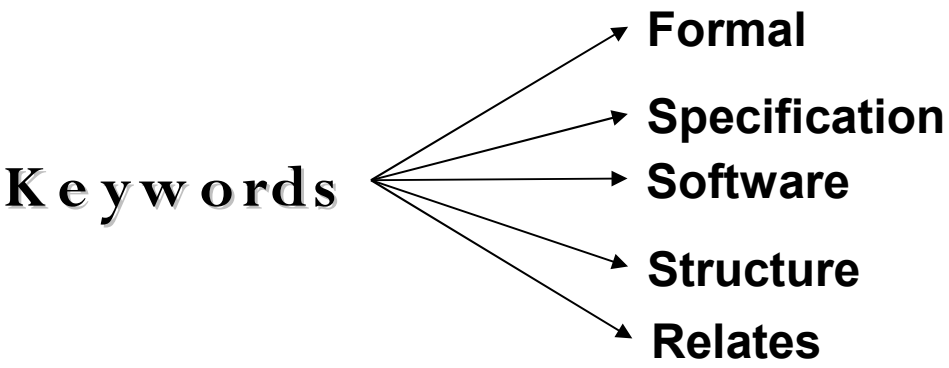


  - Reference Processes
    - **specification of common processes that are reusable across multiple projects**

  - SMART documents
    - **building codes, specifications, standards etc. tagged for use with IFC based applications**

# A Schema

- The key deliverable of the IAI is the IFC Schema.
  - this is a formal specification that can be used by software authors to create the IFC compliant software applications.
  - used to represent the structure of information and how that information relates to other information.



# What is IFC (really) ?

- IFC is not (for our interest) …
  - International Finance Corporation (www.ifc.org)
  - Inter Fraternity Council of a US based university
  - International Federation of Cheerleaders (www.**ifc-hdqrs.org**)
  - Internet Foundation Class (getting close though!! )
  - Industry Foundation Classes

- Industry Foundation Classes
  - Is this a memorable name?
  - Will you remember it tomorrow morning?
- **Think about IFC as**
  - **Information For Construction**

# IFC Challenges

- IFC is a definition of a standard format to describe a BIM
- IFC defines how information should be provided/stored for all stages of a building projects lifecycle.
- It goes right down to the properties of the individual object
- It can do  from “very little” information to “everything”.
- IFC can hold data for geometry, calculation, quantities, facility management, pricing etc. …. for many different professions (architect, electrical, HVAC, structural, terrain etc. )

# More Challenges

- If you are not specific about the information you need, how can you be certain you'll get it
  - (i.e. you could get a terrain model for a electrical short circuit calculation)
- Will different users produce the same model with different software ?
- Will different users produce the same model with the same software...
- Is it enough to ask for IFC files in a project ?

# The Example

- The following set of slides will use a ‘Fan’ as an example
- A fan is an air pump that creates a pressure difference and causes airflow.

- It is NOT ‘someone who has an intense, occasionally overwhelming liking of a person or group of persons’

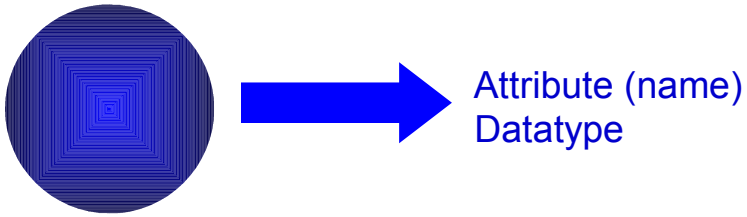
*Star Trek Fan !!!!*

# IFC Schema Representations

- Iconic (Graphical)
  - The iconic (graphical) notation used to create the IFC schema is known as EXPRESS-G.
  - It makes model development easier
  - It makes model review easier.
- Formal Specification
  - This formal specification of the schema uses the international standard data definition language known as EXPRESS.
  - A version of the specification also exists in XML (called ifcXML)

# Classes and Objects

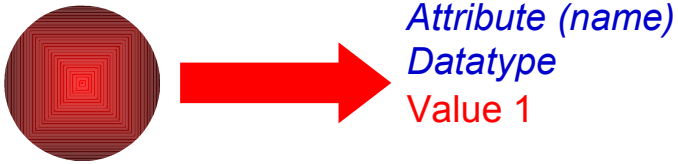
**Specification of Thing (Class)**



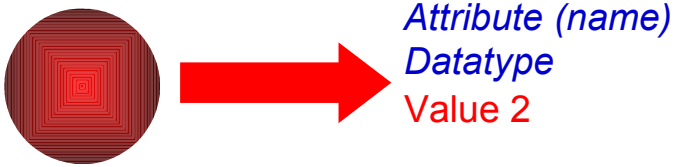

**Fan**

**Vifte (Norwegian)**
**Ventilateur (French)**
**Anhänger (German)**
**Ventilador (Spanish)**

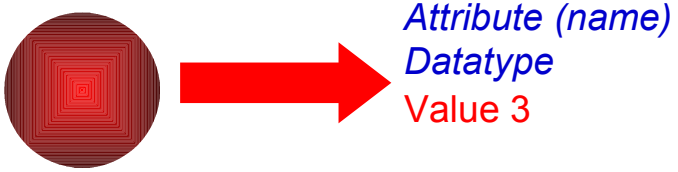
**Things that exist (Objects)**




**Fan 1**

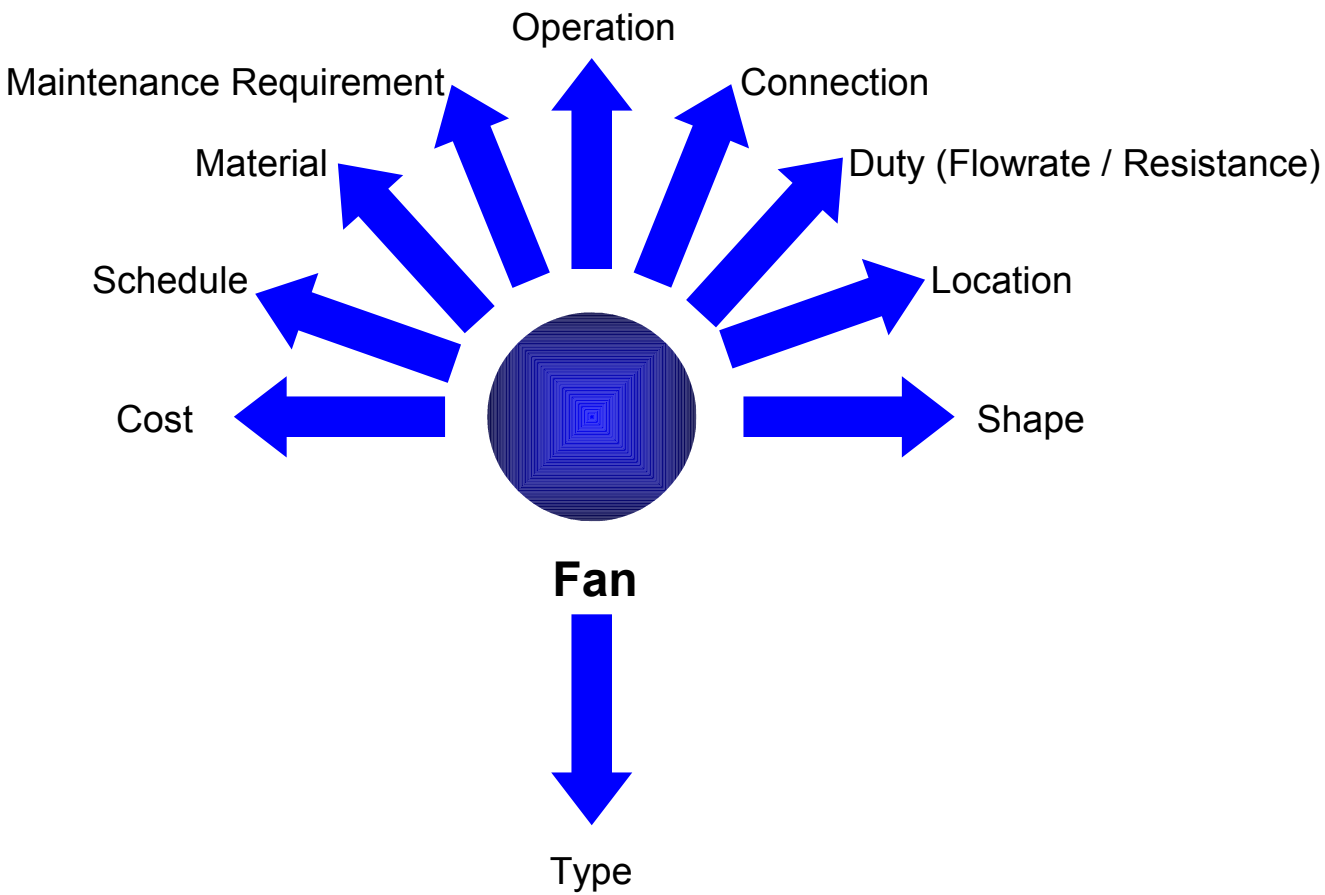



**Fan 2**

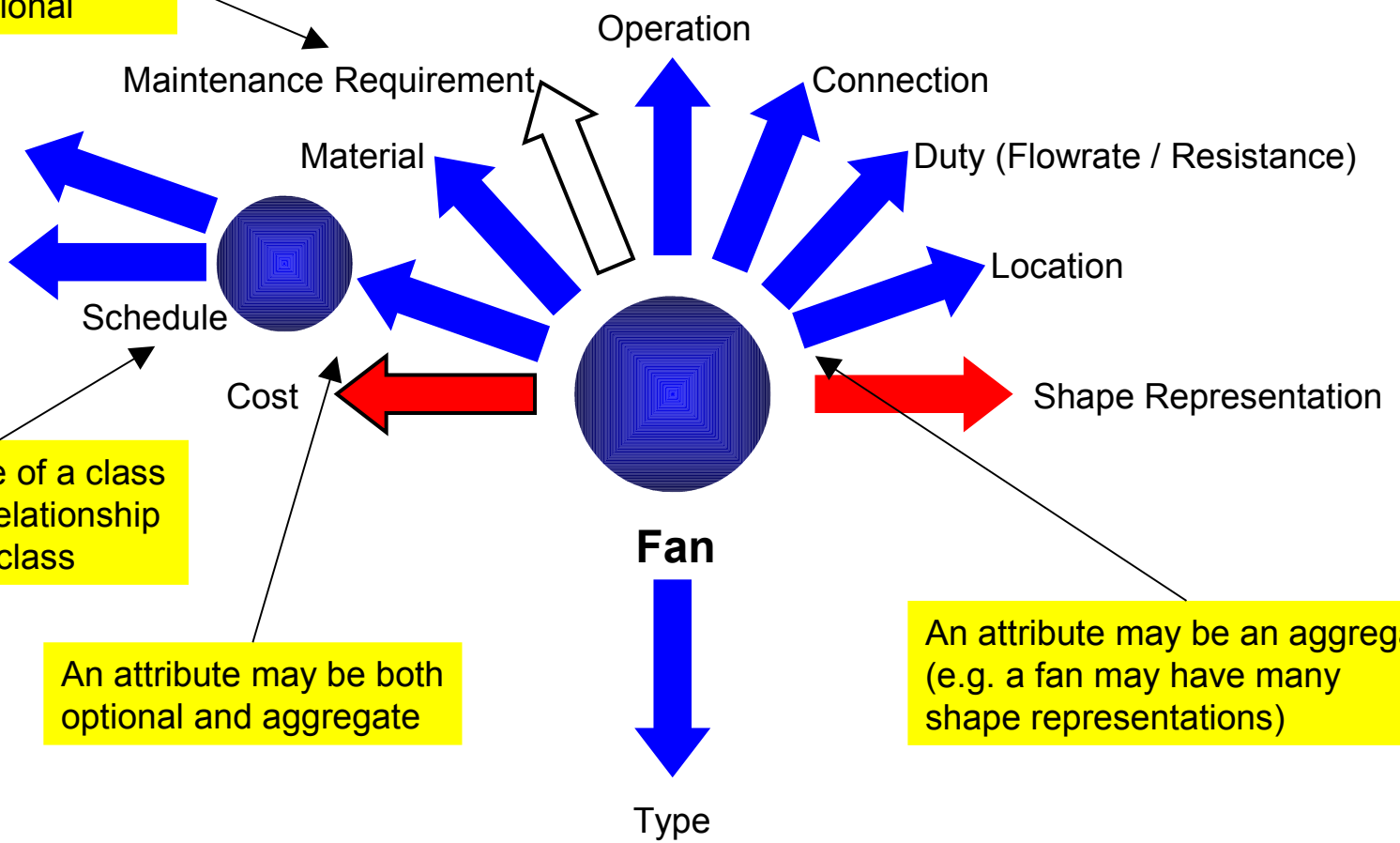



**Fan 3**

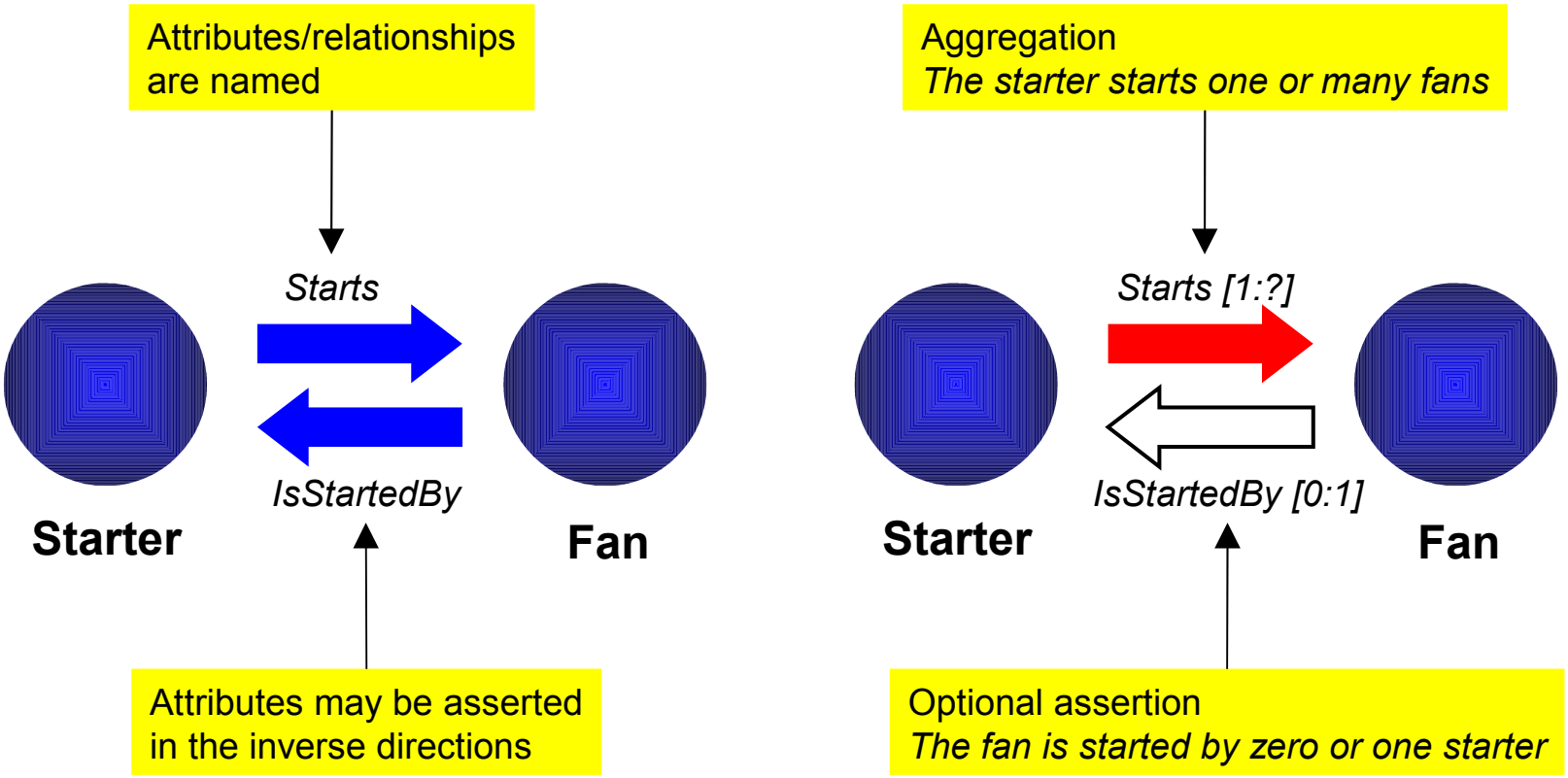
# Attributes


Operation
Maintenance Requirement
Connection
Material
Duty (Flowrate / Resistance)
Schedule
Location
Cost
Shape
Fan
Type

# Attributes

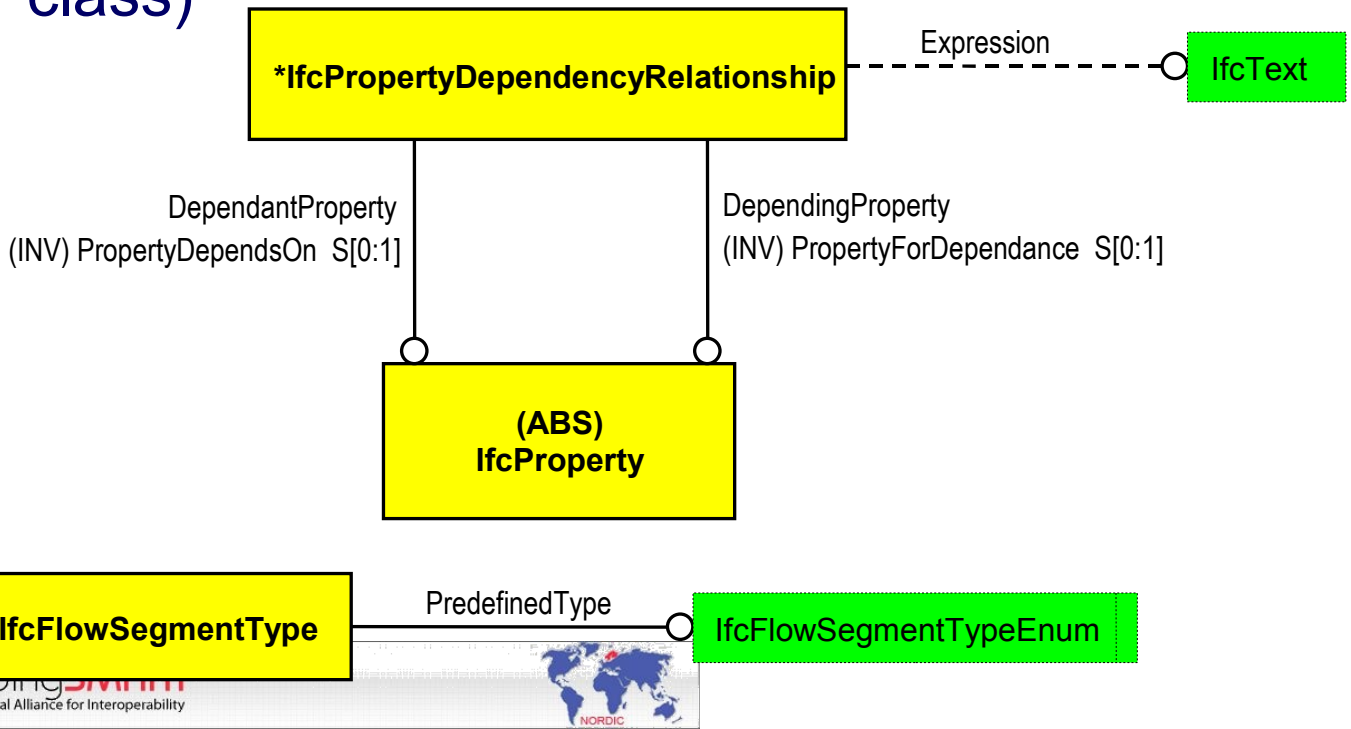

Asserting an attribute may be optional
Maintenance Requirement
Operation
Connection
Material
Duty (Flowrate / Resistance)
Location
Schedule
Cost
Shape Representation
Fan
Type
An attribute of a class may be a relationship to another class
An attribute may be both optional and aggregate
An attribute may be an aggregate (e.g. a fan may have many shape representations)

# Inverse attributes



# Iconic Schema

- Classes and attributes are specified using graphical ‘icons’
- A class is shown in a box
- An attribute is shown as a line with an end marker
- The end marker connects to a datatype (which may be another class)



# Schema Specification

- Iconic representation of the schema is then converted to a formal language
- This can be understood and checked by computers
- It is for data exchange and database design
  - not for processing like VB, C++, Java etc.

```
ENTITY IfcPropertyDependencyRelationship;
        DependingProperty : IfcProperty;
        DependantProperty : IfcProperty;
 WHERE
        WR1 : DependingProperty :<>: DependantProperty;
END_ENTITY;
```

# Exchange Files

- The formal object model is a specification of how data is structured in a file
  - if the object model is a public specification
  - if system A can export data according to the specification
  - if system B can import data according to the specification
  - this is the basis for a neutral format standard
- Neutral
  - in the public domain
  - not the property of any software vendor

```
#10 IfcPropertyDependencyRelationship (#20, #30);
#20 IfcProperty (….);
#30 IfcProperty (….);
```

# Documentation

- The IFC schema is available via the IAI web site
- The whole model can be seen through a browser
  - IE7, Firefox, Netscape, Opera
- Every class and attribute has its meaning defined

**IfcPropertyDependencyRelationship**

| | |
|---|---|
| Definition from IAI: | An IfcPropertyDependencyRelationship describes an identified dependency between the value of one property and that of another. |
| Use Definition: | Whilst the IfcPropertyDependencyRelationship may be used to describe the dependency, and it may do so in terms of the expression of how the dependency operates, it is not possible through the current IFC model for the value of the related property to be actually derived from the value of the relating property. The determination of value according to the dependency is required to be performed by an application that can then use the Expression attribute to flag the form of the dependency. |

*Attributes*

| | |
|---|---|
| DependingProperty: | The property on which the relationship depends. |
| DependantProperty: | The dependant property. |

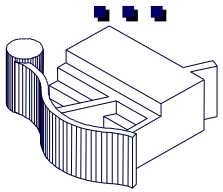
# User Documents

- There are also a number of guidance documents available on the web site including:
  - IFC Model Extension Guide
    - Provides some more detailed information on how to build an IFC model
  - IFC Model Integration Guide
    - Shows how models from different IAI development projects are brought to a single model
  - IFC Implementation Guide
    - Provides extensive information and examples to software developers wishing to implement IFC
  - Readers Guides
    - EXPRESS-G
    - EXPRESS
    - Converting EXPRESS to IFC File Structure

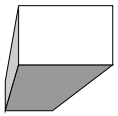
buildingSMART

# can

## Shape (explicit)

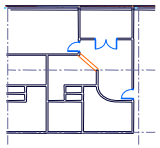


## Shape (extrusions)
beams, pipes, ducts, walls etc.

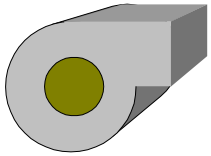


## Shape (topology)
line representations for pipe, duct, etc.

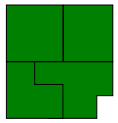
## Building Elements
wall, door, window, roof, stairs, etc.



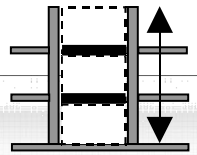
## HVAC Equipment
chillers, fans, pumps, boilers, coils, cooling towers, heaters, heat exchangers, etc..

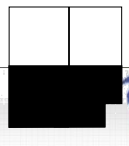


## Spaces, Space Structure
space, storey, part, building, site

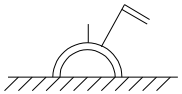


## Zones, Compartments
fire, workstation, rising ducts, shafts



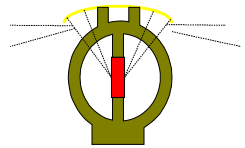


## Electrical Elements
transformers, motors, generators, switches, protective devices, power and communication outlets panels, cubicles

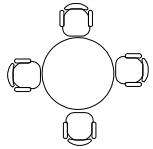


## Sanitary Elements
WC's, urinals, baths, bidets, traps gulleys

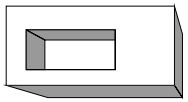
## Fire Protection Elements
sprinklers, hose reels, hydrants, wet/dry rising mains



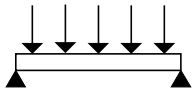
## Furniture
inc. system furniture



## Relations Between Elements
holes, chases, voids, zones



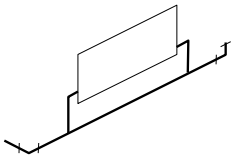
## Structural analysis:
structural members, boundary conditions, connections, supports, loads, etc.



## Structural Elements
members, profiles, rebars, properties, joints, features, surface

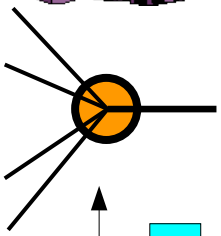
# and ...

## Systems
piping, ducting, cable, structural

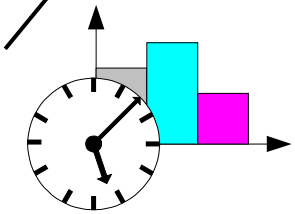


## Lighting
fittings, rendering, photo-accurate lighting

## Manholes
manholes, inspection chambers, access chambers, meter chambers, valve chambers



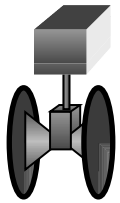
## Time Series
time related events



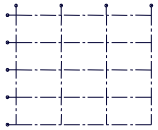
## Constraints
rules, specifications, requirements trigger conditions

## Environmental Impact
embodied energy, $CO_2$

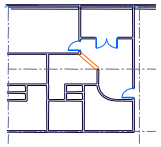
## Controls, Instruments
sensor, actuator, controller, gauge, meter



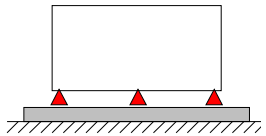
## Grids



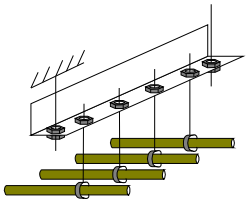
## Draughting



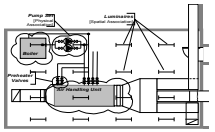
## Holes and Bases
holes, sleeves, packing, framing, upstands, vibration isolation



## Accessories
brackets, drop rods, steel sections, bracket assemblies, screws, bolts etc.



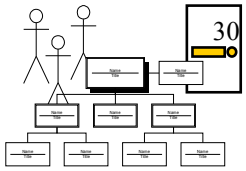
## Asset Management
maintenance history, inventories



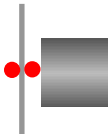
## Help
request, action, permit, warranty, operation

# ...and

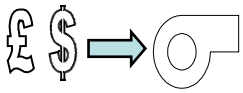
**Actors**
people, organizations, addresses



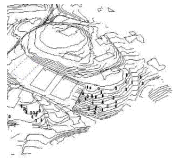

**Connectivity**
services, structure, building



**Costing**
cost planning, estimates, budgets, whole life

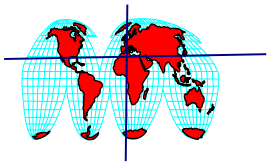



**Geographical Elements**
features, contours, regions

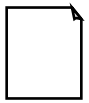


**Work Plans and Schedules**
inc. nested schedules, resource allocation

**Coordinate Mapping**
geodetic, cartesian

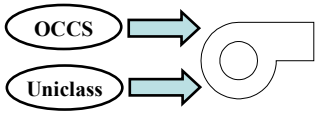


**Orders**
work orders, change orders, purchase orders

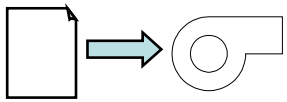


**External Data**

**Classification**



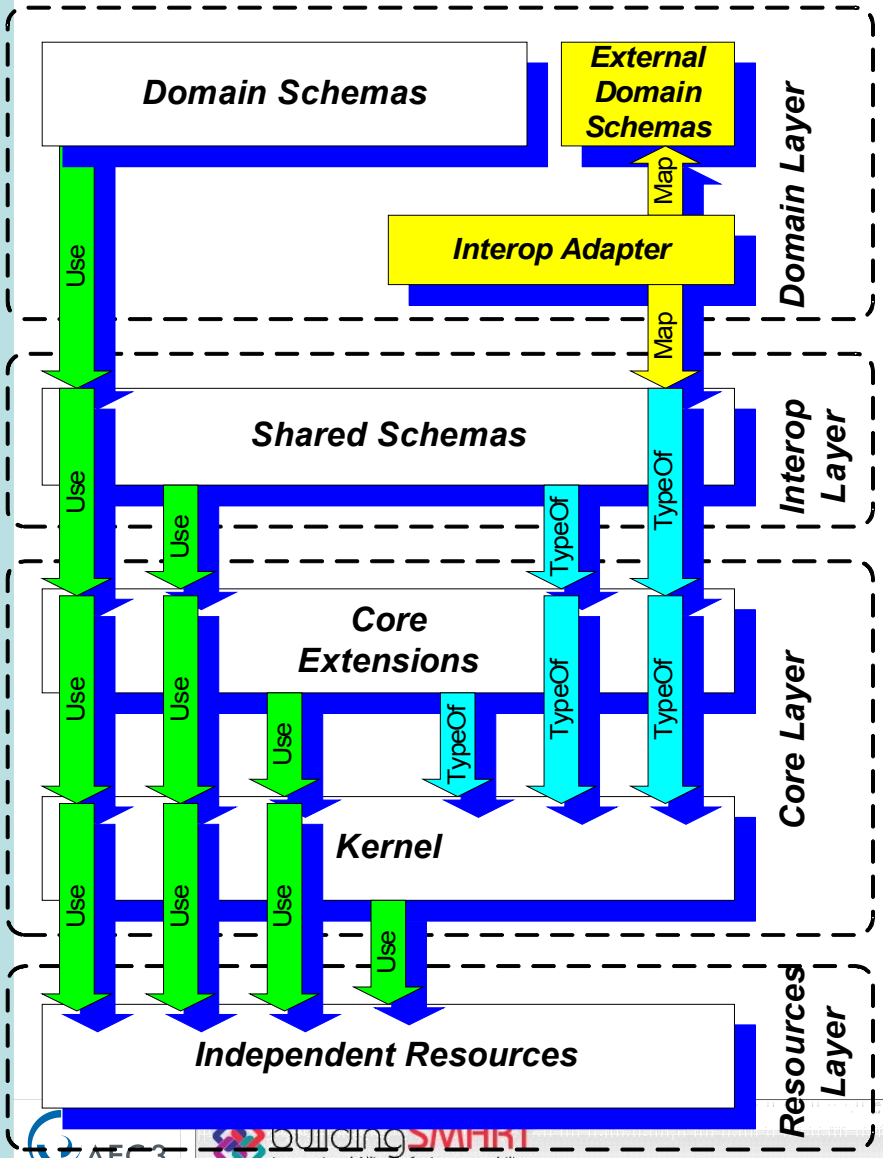

**Associated Documents**



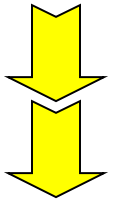
# Creating the IFC Project Model

Jeffrey Wix

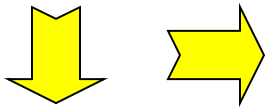
# IFC Schema Definition Layers



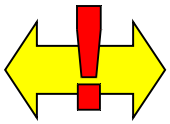

The IFC schema operates on a 'ladder principle'.



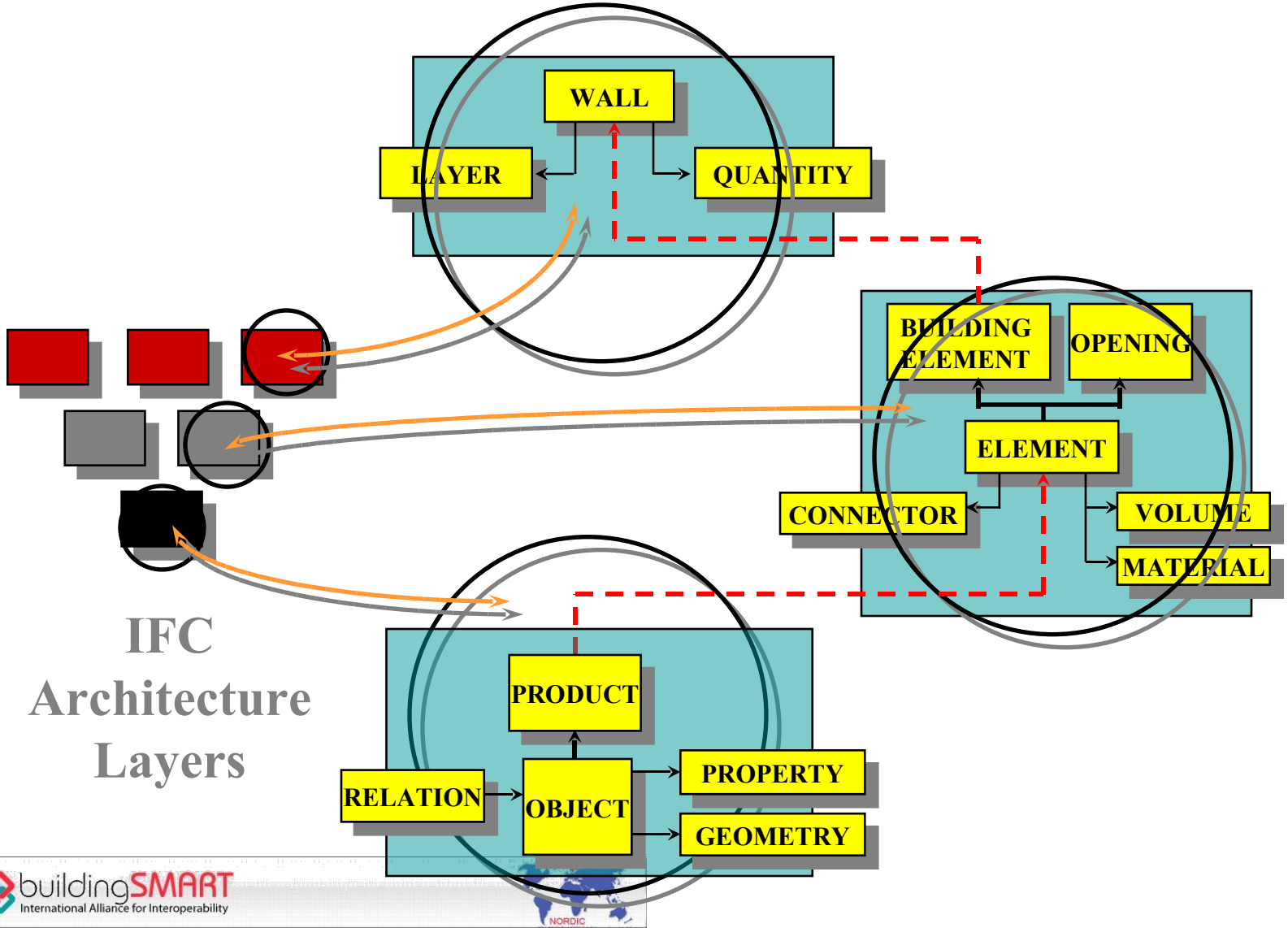
Any class may reference a class at the same or lower layer but may not reference a class from a higher layer.



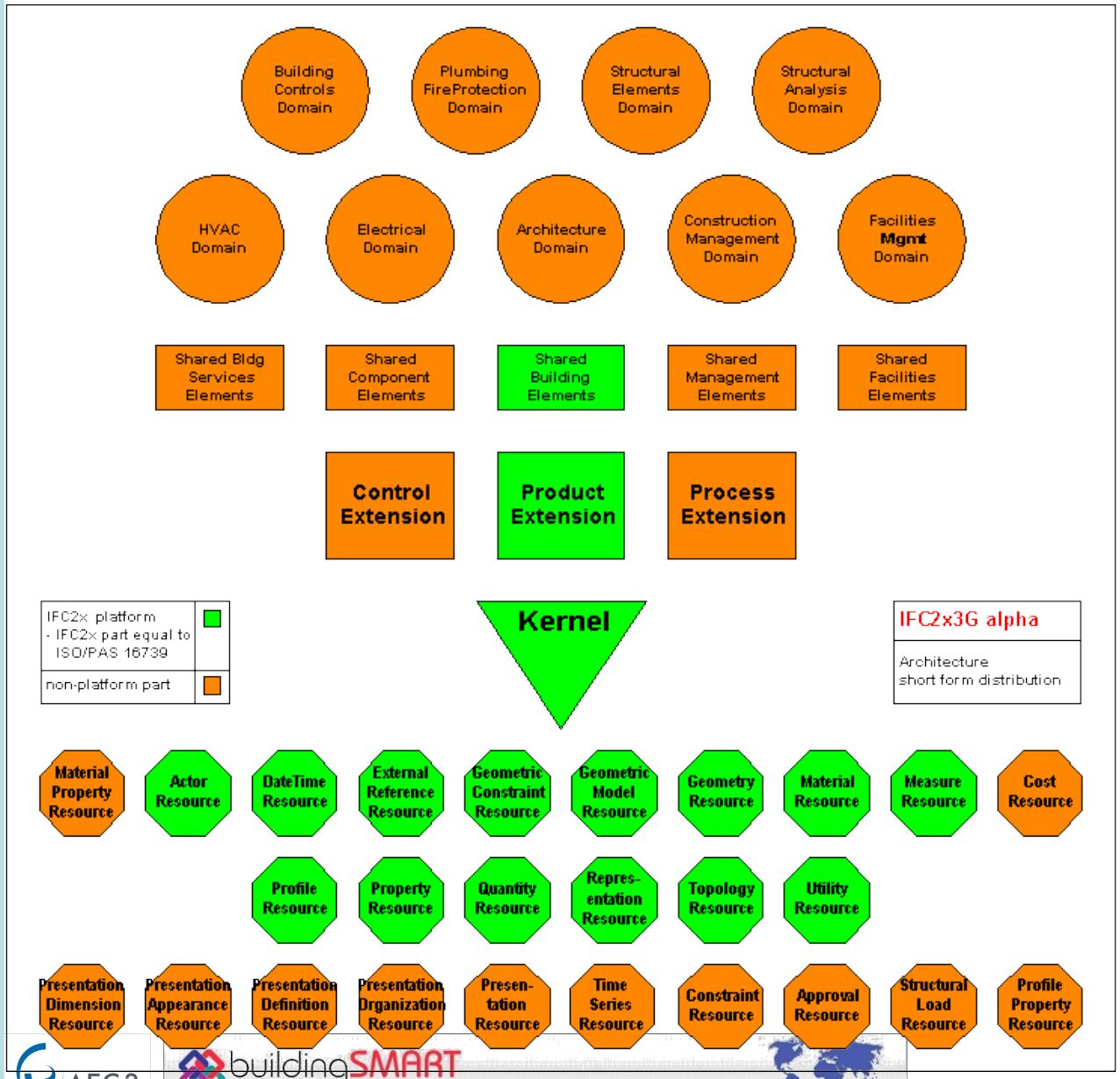
References within same layer must be designed very carefully and are only allowed within the Core layer and the Resource layer.



# Definition Levels



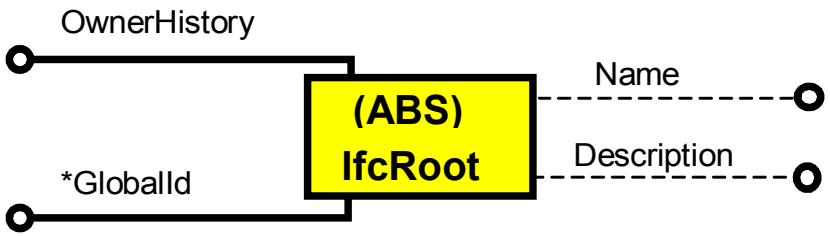
# Schemas




- IFC comprises a set of schemas
- Each schema belongs to one IFC layer
- The content of a schema details a particular ‘idea’ (cost, process, geometry, HVAC etc.)

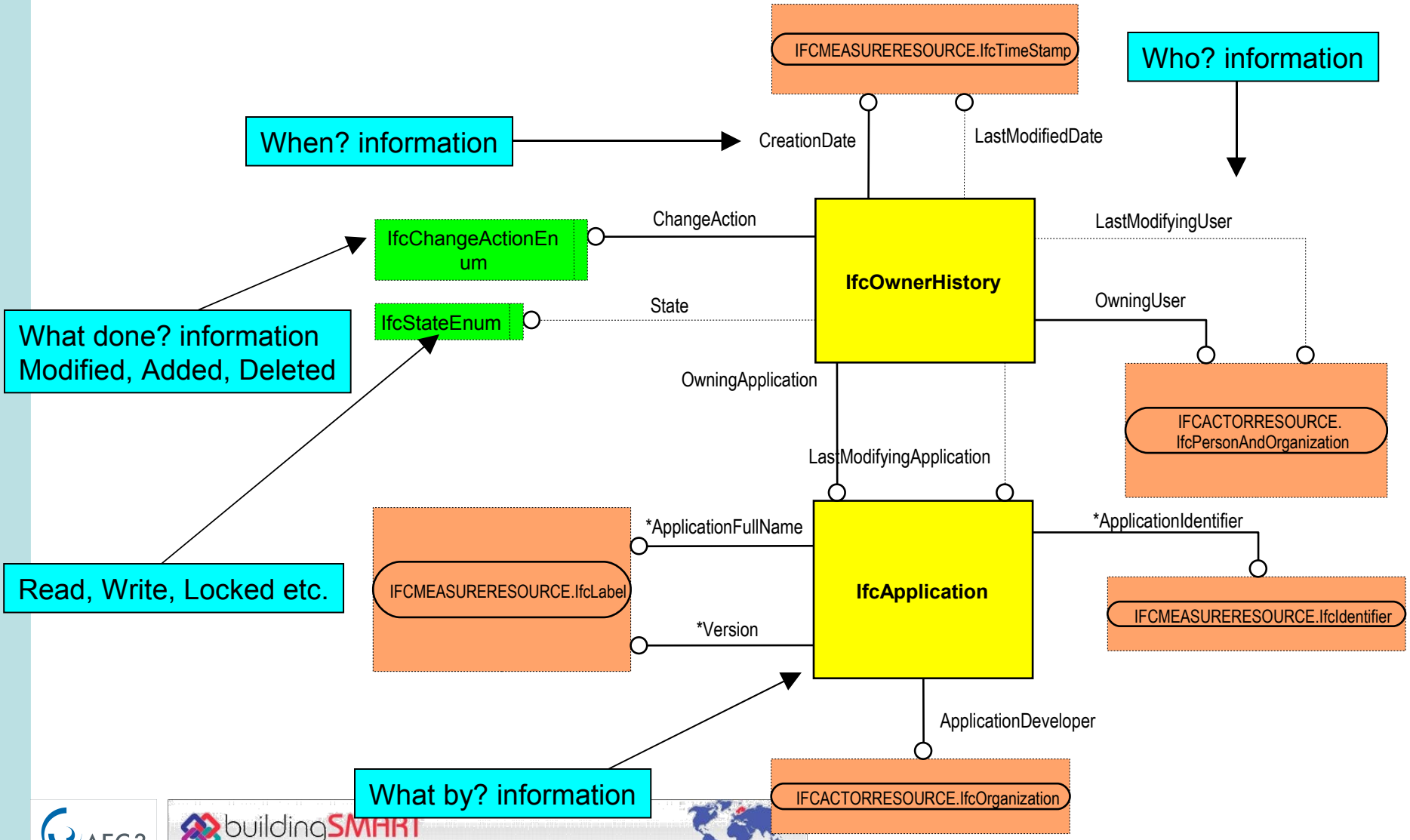
# The ‘Root’ class

- All classes in IFC, except resource classes, are subtypes of the **IfcRoot** class
- IfcRoot provides basic services for the model
  - Object identity
  - Local naming and description
  - Ownership and change information



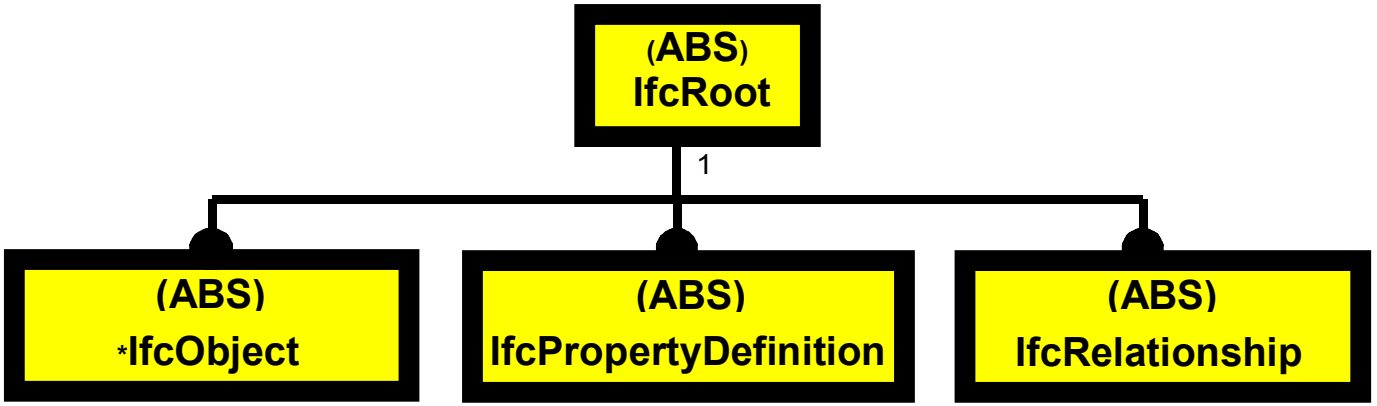
# Owner History

- Not mandatory, but recommended



# Fundamental Classes

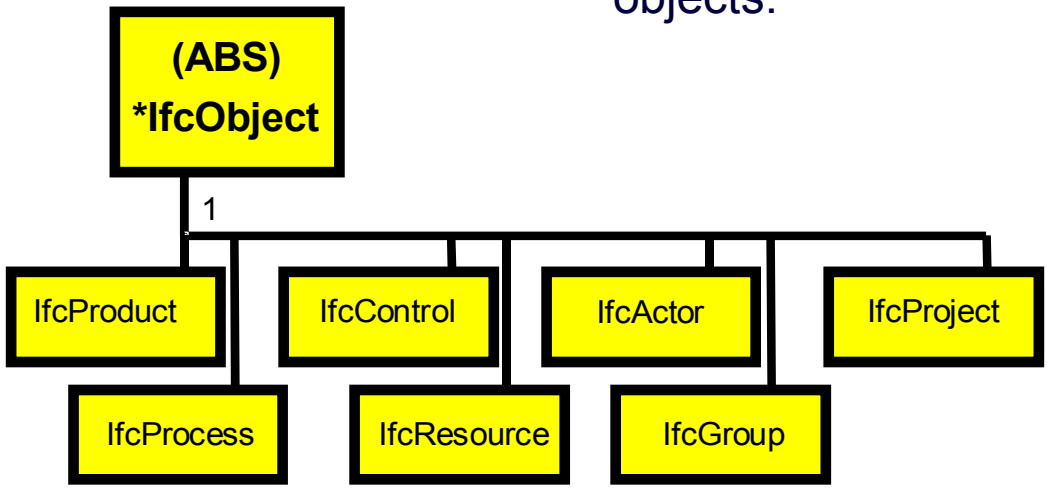
- There are three fundamental classes in the IFC model, which are all derived from **IfcRoot**.
  - objects are the generalization of any thing (or item)
  - relations are the generalization of all relationships among things (or items)
  - properties are the generalization of all characteristics (either types or partial type, i.e. property sets) that may be assigned to objects



# Objects

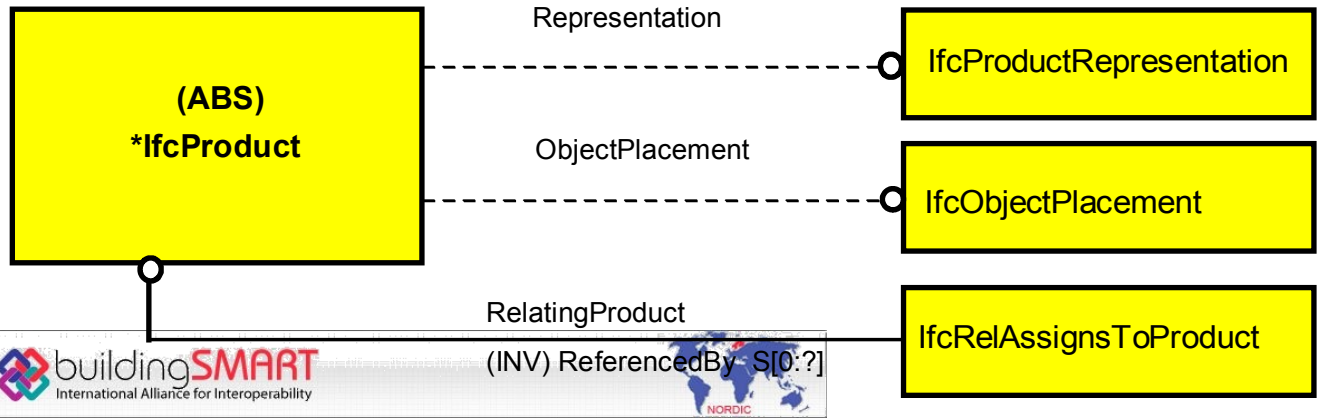
- There are seven fundamental types of **IfcObject**.

- **Products** are physical things (manufactured, supplied or created).
- **Processes** are actions of, e.g., acquiring, constructing, or maintaining
- **Controls** are concepts that constrain other objects e.g. guide, specification, regulation, that has to be fulfilled.
- **Resources** are concepts describing the use of an object within a process
- **Actors** are human or organizational agents
- **Project** is the undertaking of activities leading towards a product.
- **Group** is an arbitrary collection of objects.



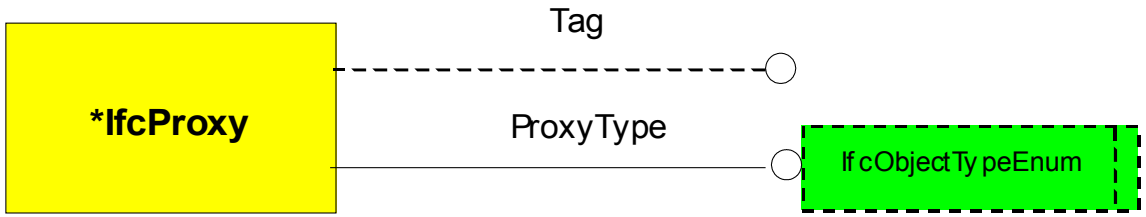
# Product

- **IfcProduct** handles physical items incorporated into a project either as supplied or by construction/assembly of other products
  - ObjectPlacement can be:
    - absolute (relative to the world coordinate system)
    - relative (relative to the object placement of another product)
    - constrained (e.g. relative to grid axes).
  - Representation is the container of potentially multiple geometric representations.
    - All geometric representations are defined within the ObjectPlacement, which provides the object coordinate system.



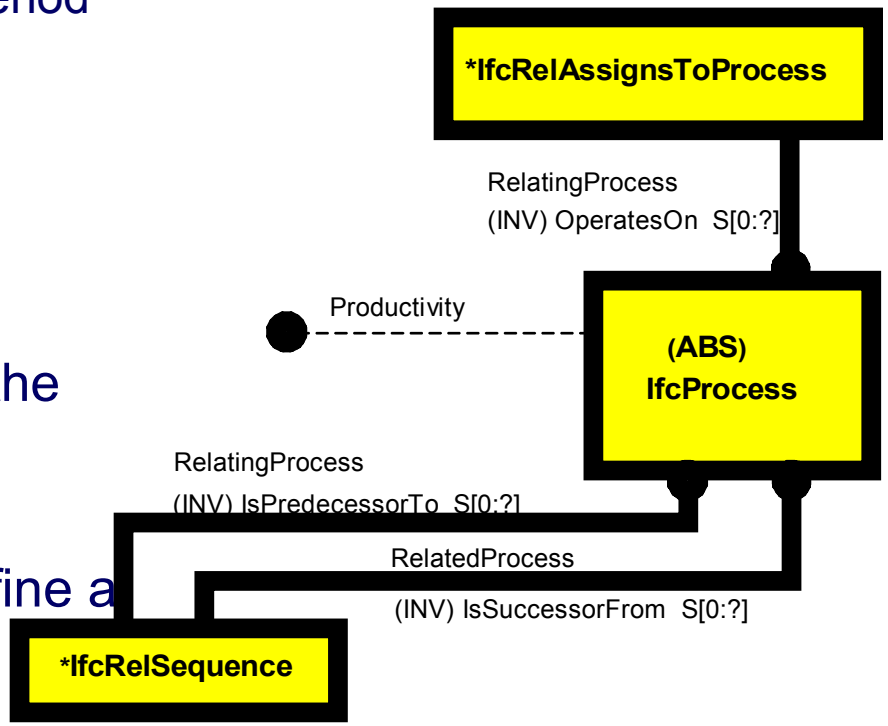
# Proxy

- **IfcProxy** is a special subtype of product used for objects not otherwise in the IFC model.
  - A proxy may have a representation and placement (inherited from *IfcProduct*)
  - It can be further defined by property definitions



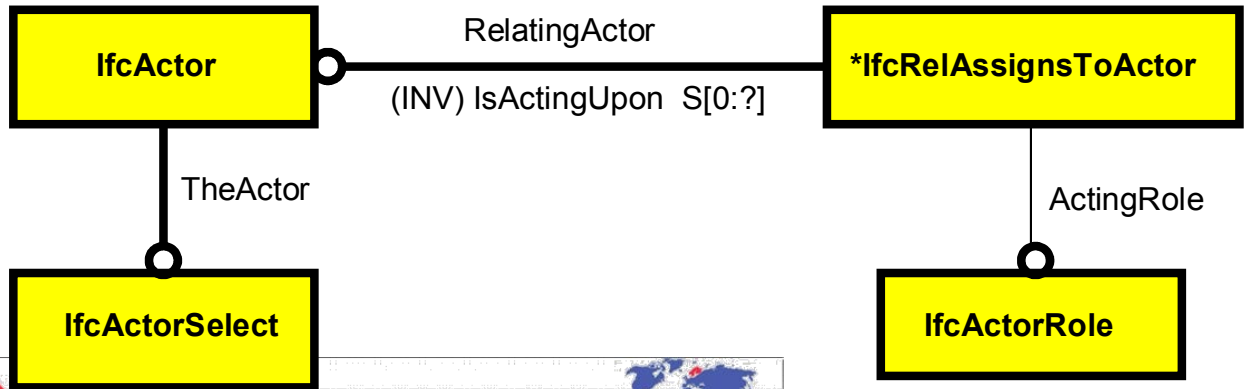
# Process

- **IfcProcess** captures activities
  - work being carried out over a period of time.
  - processes may be named and described
  - may incorporate a measure of productivity
- Relates other objects on which the process operates on as input or output
- Predecessor and Successor define a sequence relationship



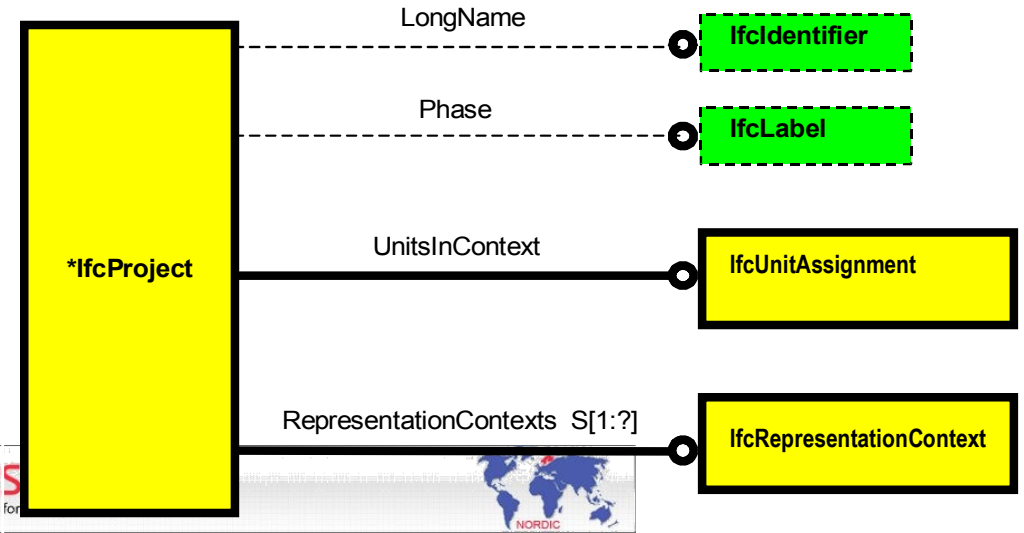
# Actor

- **IfcActor** handles people and organizations active within a project.
  - Captures information about
    - participants within a project
    - future occupants of the building within an FM context.
  - Actors can have names, addresses, affiliations, roles, assignments to objects

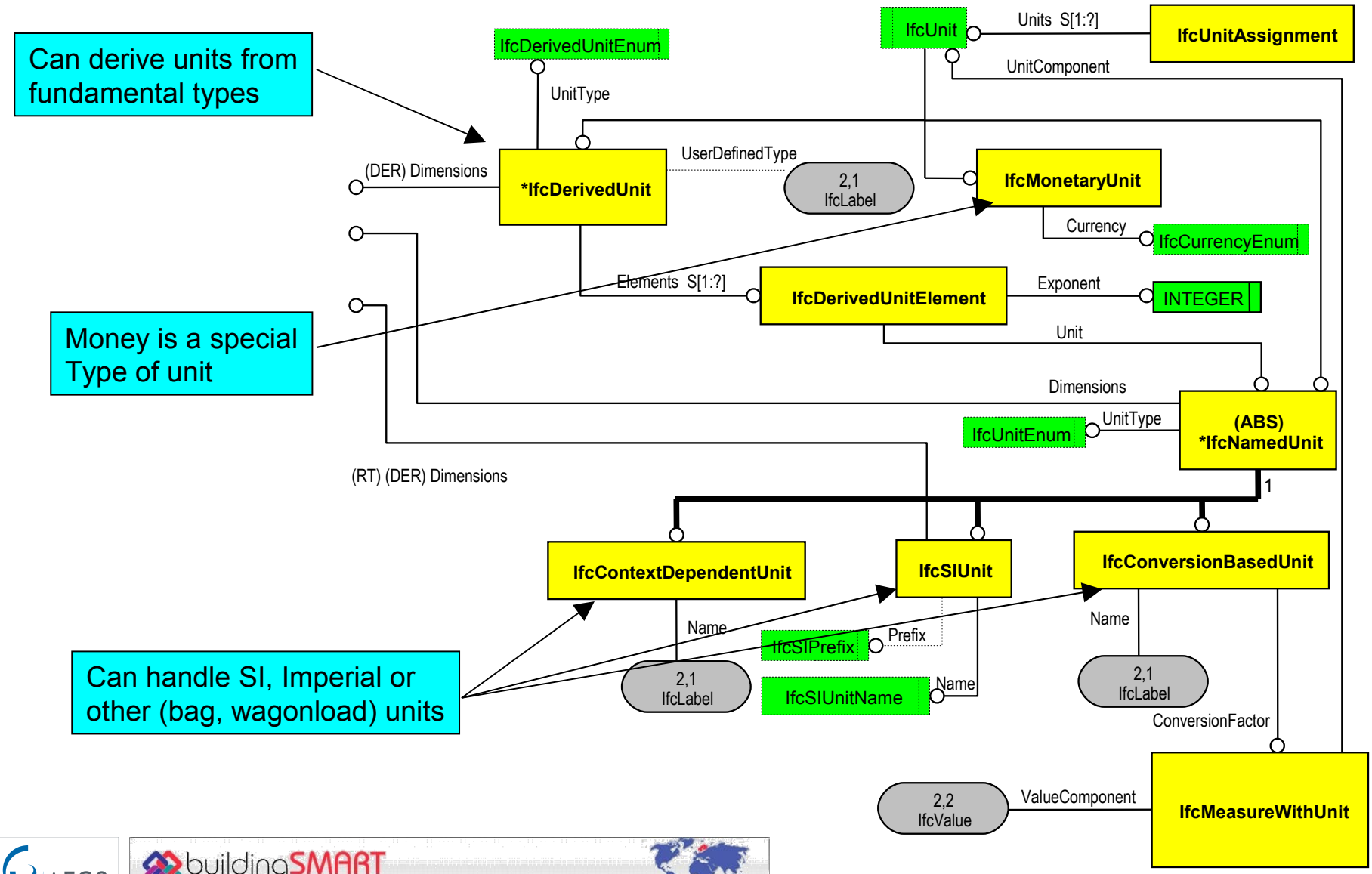


# Project

- **IfcProject** is the container for all information exchanged
  - There can only be one IfcProject within an IFC file or database
  - IfcProject holds global information about the project (that can only be declared once) including:
    - the default units used
    - the world coordinate system
    - the coordinate space dimension
    - the precision used within the geometric representations
    - indication of the true north relative to the world coordinate system

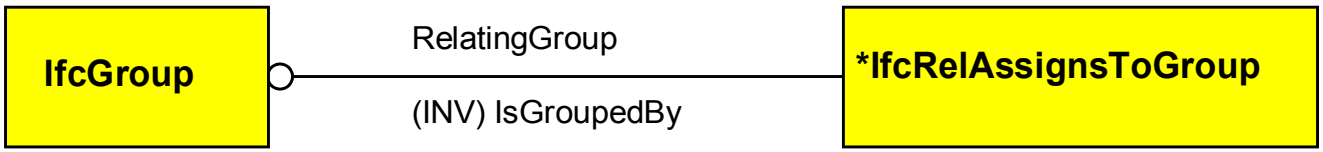


# Units


Can derive units from fundamental types
Money is a special Type of unit
Can handle SI, Imperial or other (bag, wagonload) units
IfcDerivedUnitEnum
UnitType
*IfcDerivedUnit
(DER) Dimensions
UserDefinedType
2,1 IfcLabel
IfcUnit
Units S[1:?]
IfcUnitAssignment
UnitComponent
IfcMonetaryUnit
Currency
IfcCurrencyEnum
Elements S[1:?]
IfcDerivedUnitElement
Exponent
INTEGER
Unit
Dimensions
IfcUnitEnum
UnitType
(ABS) *IfcNamedUnit
(RT) (DER) Dimensions
1
IfcContextDependentUnit
IfcSIUnit
IfcConversionBasedUnit
Name
2,1 IfcLabel
IfcSIPrefix
Prefix
IfcSIUnitName
Name
Name
2,1 IfcLabel
ConversionFactor
2,2 IfcValue
ValueComponent
IfcMeasureWithUnit

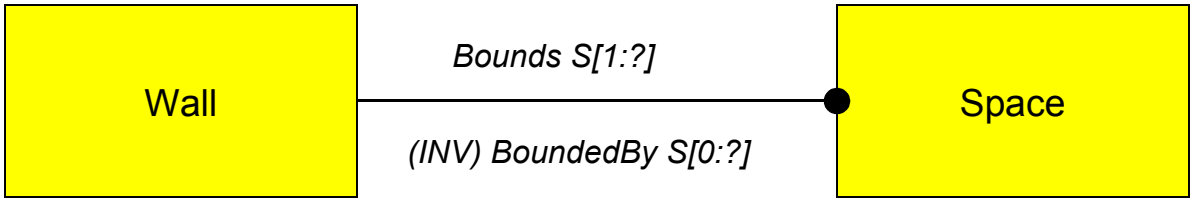
# Group

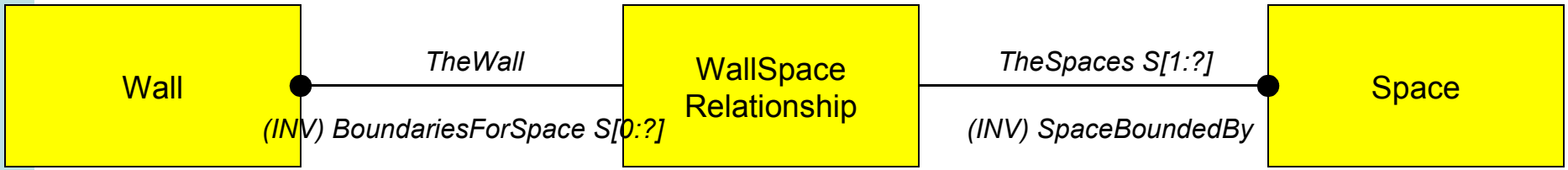
- **IfcGroup** brings together other objects so that they can be considered as a logical unity within a project
- A single object may be included within several groups.
- A group may be part of another group.
  - but cannot be part of itself



# Relationships

- Class to class relationships are handled by turning the relationship into an object
- This is called an *objectified relationship*

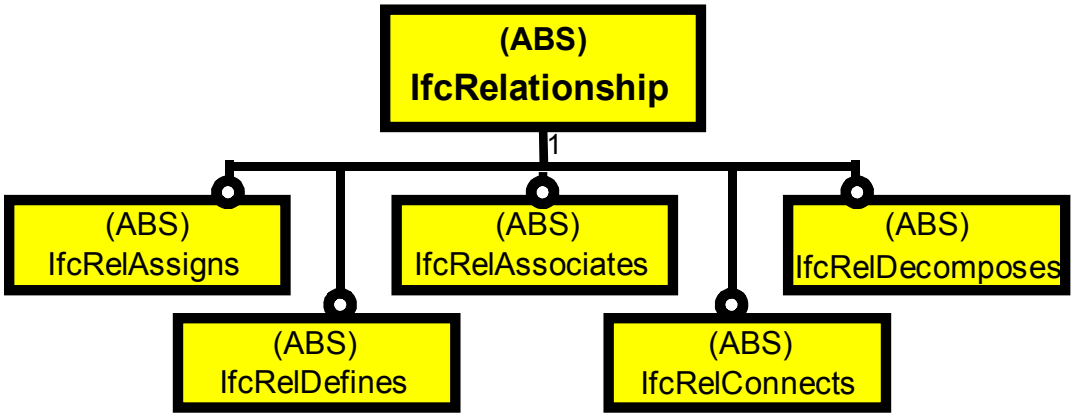


# Relationship Types

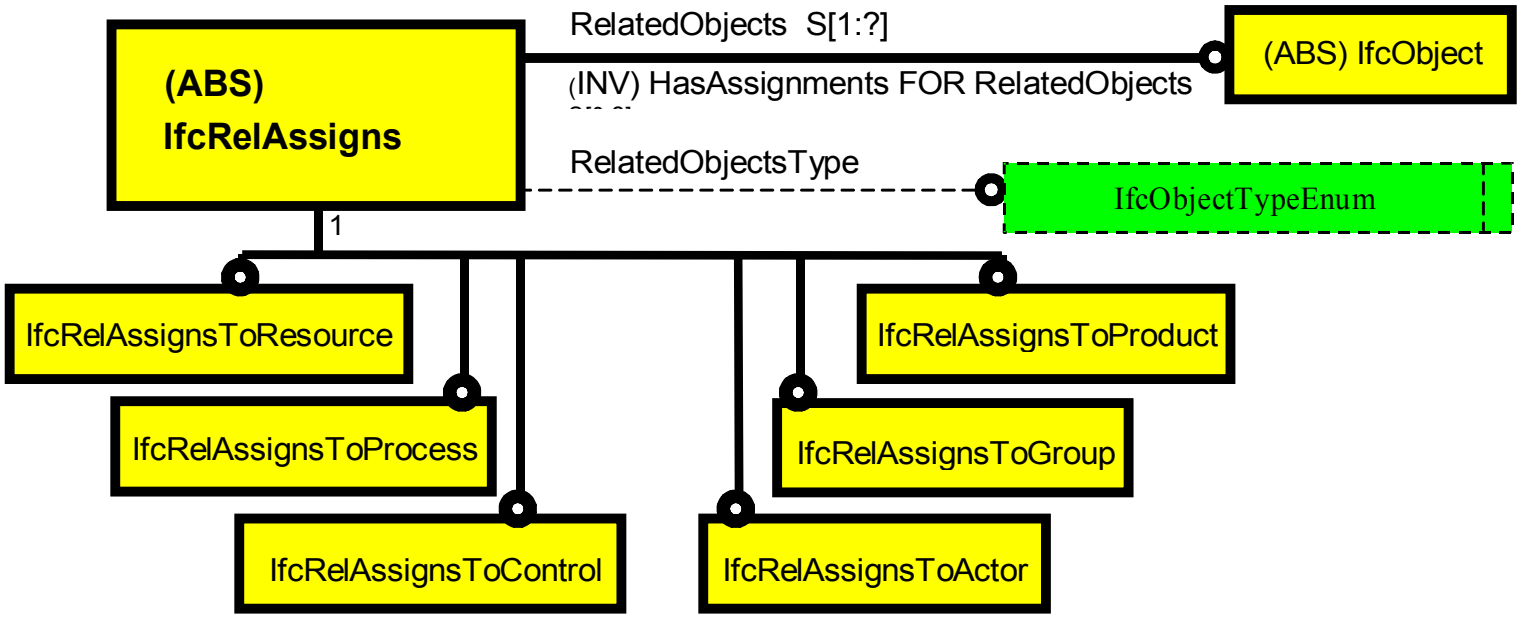
- There are five fundamental relationship types in the IFC model

- **Assignment** –enables a client object to apply the services of other objects
- **Association** – enables external sources (classification, library or document) to be associated with objects or property definitions.
- **Decomposition** –defines a whole/part hierarchy or assembly.
- **Definition** – applies a type definition or property set definition (seen as partial type information) to an object.
- **Connection** – defines the connectivity between objects

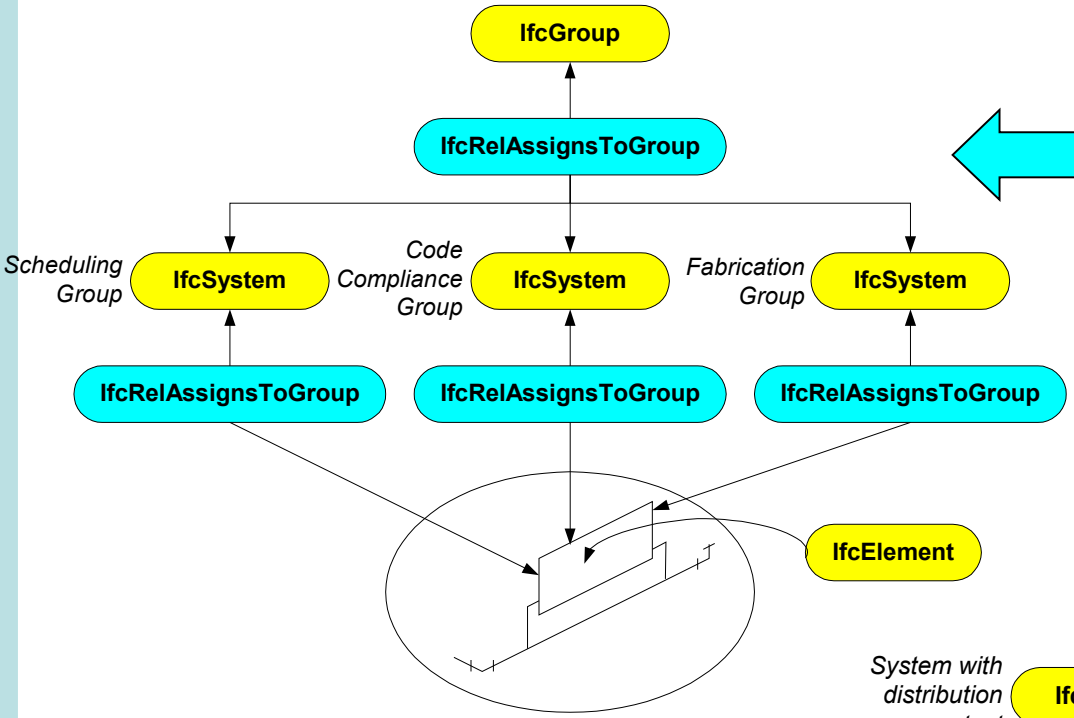


# Assignment

- This is the most general form of relationship in IFC
- It allows relationships to be made from any object to any other that is a main IfcObject subtype

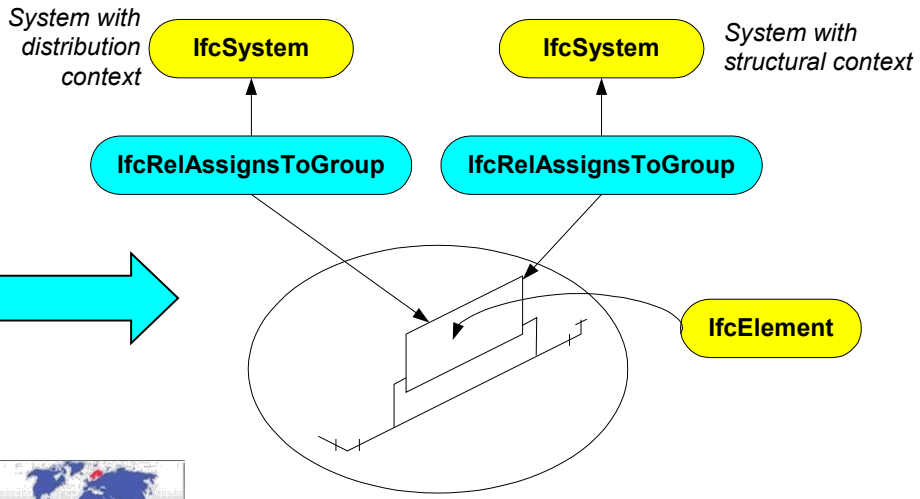


# Assignment to a ‘System’ Group



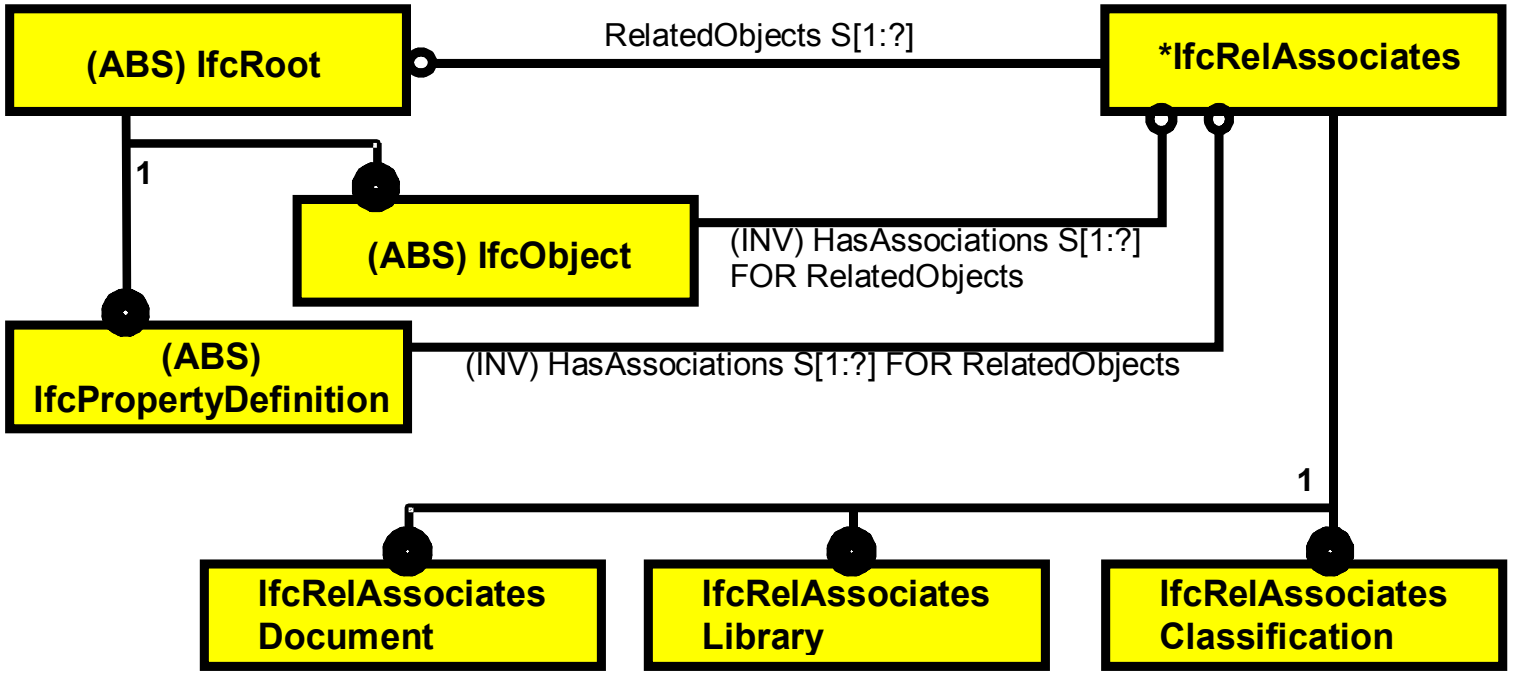

Assigning an element to different ‘purpose’ systems

Assigning an element to different ‘functional’ systems



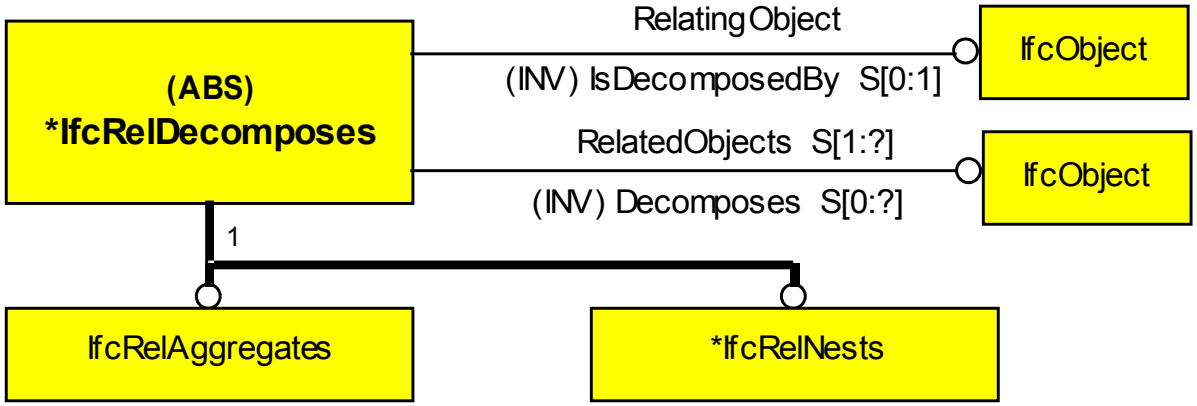
# Association

- An object may have an association to a document (paper, electronic etc.)
  - The pointer is to the document as a container and not to the document information structure
- An object may have one or many classifications (SfB, CAWS, Uniclass etc)



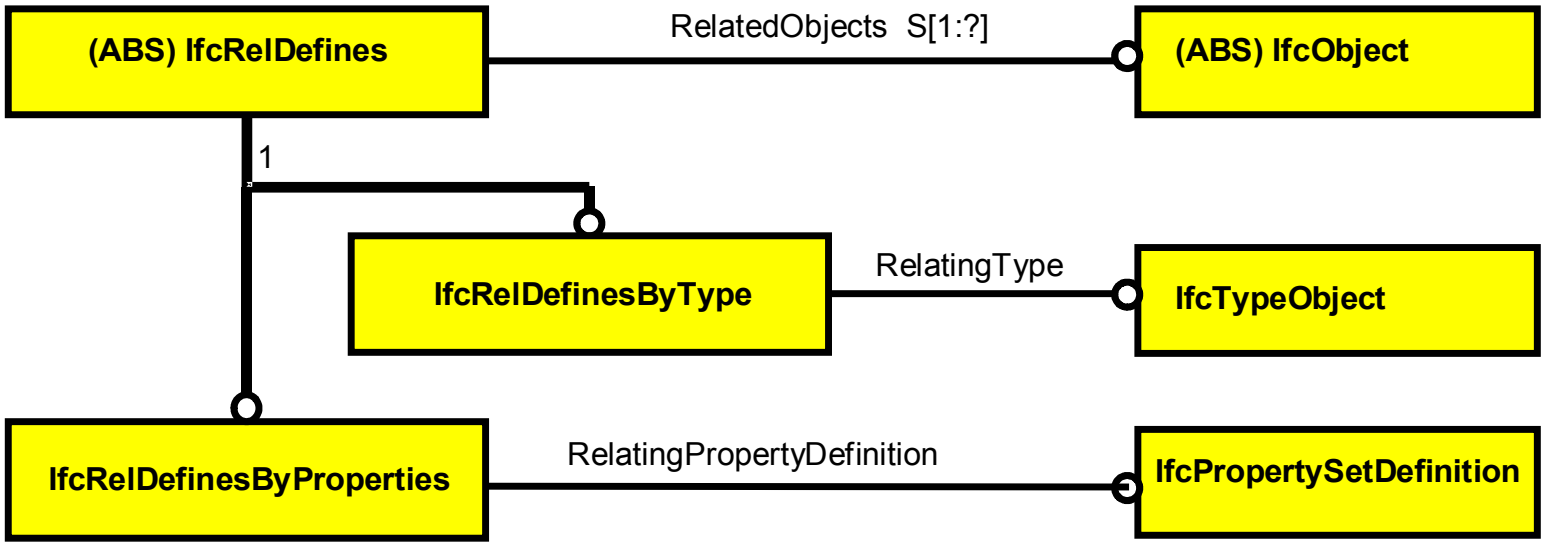
# Decomposition

- Shows an assembly or whole/part relationship
- Nesting decomposes objects of the same class
  - E.g. decomposing a system (instance of IfcSystem) into subsystems (instances of IfcSystem)
- Aggregation decomposes any type of objects



# Definition

- Used for the property set and/or type object definitions for an object.

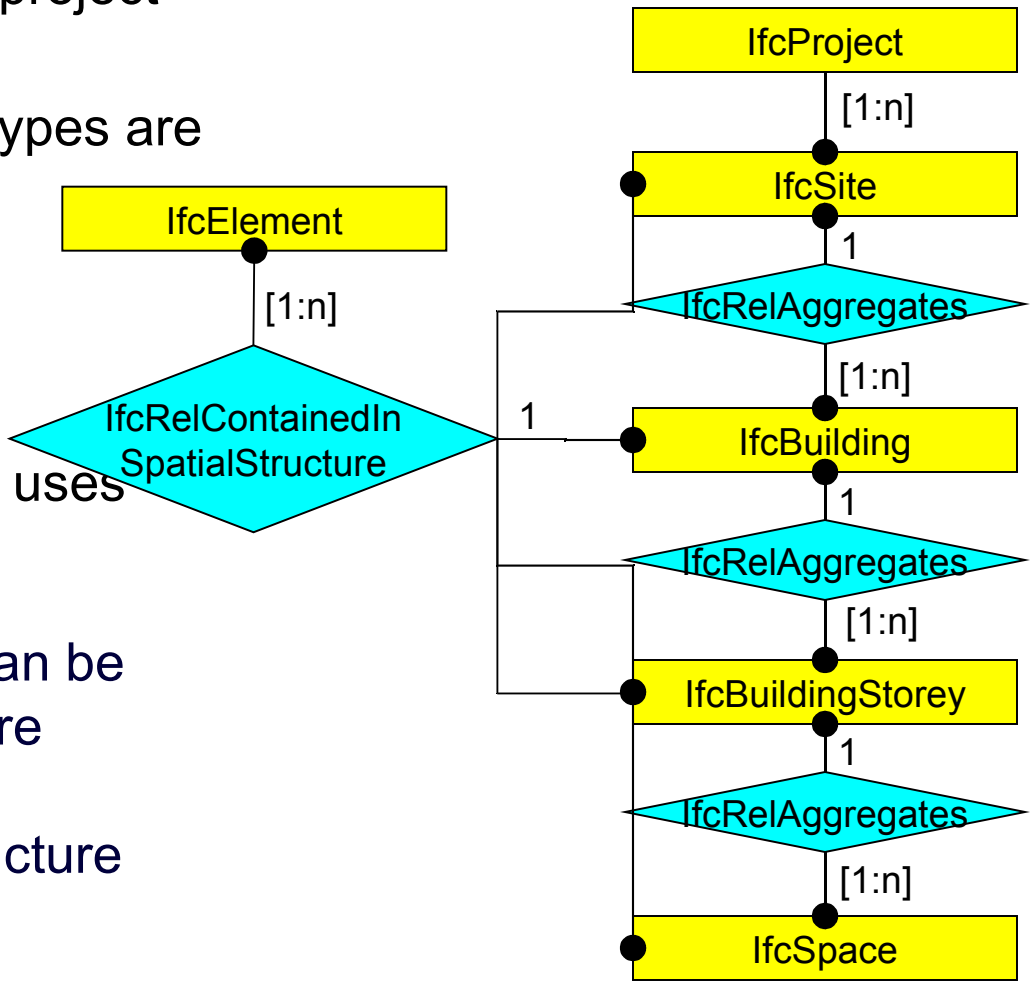


# Connection

- A connection relationship may be physical or logical.
- IfcRelConnects is always elaborated by the semantics of its use which includes
  - **IfcRelVoidsElement** defines the voiding connection between an opening and a physical item
  - **IfcRelFillsElement** defines the filling connection between a physical item and an opening
  - **IfcRelContainedInSpatialStructure** defines the hierarchical connection between different spatial structures (site, building, storey, space)
  - **IfcRelSpaceBoundary** defines the connection between a space and a physical boundary item
  - **IfcRelConnectsElements** defines the physical connection between items such as walls
  - **IfcRelConnectsPorts** defines the logical connection of building services items at ports.

# Spatial Structure

- Spatial structure breaks a project into manageable subsets
- Spatial structure element types are
  - Site
  - Building
  - Building storey
  - Space
- Spatial structure hierarchy uses IfcRelAggregates



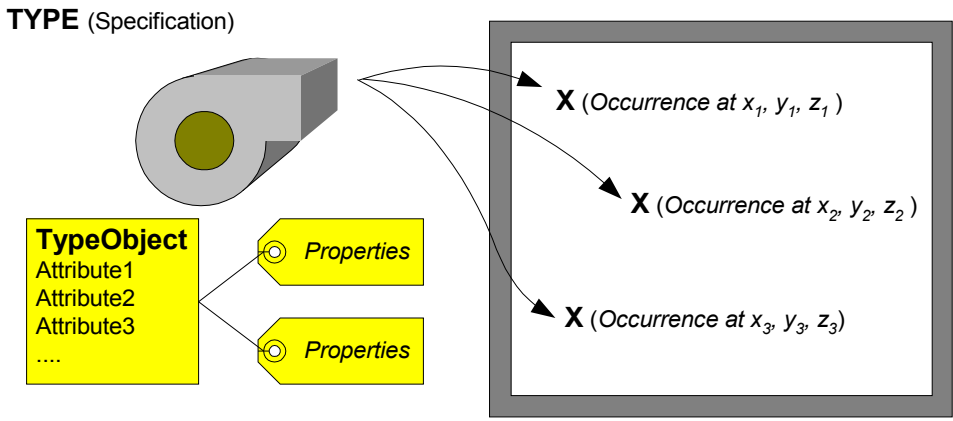

- Any instance of IfcElement can be contained in a spatial structure through IfcRelContainedInSpatialStructure

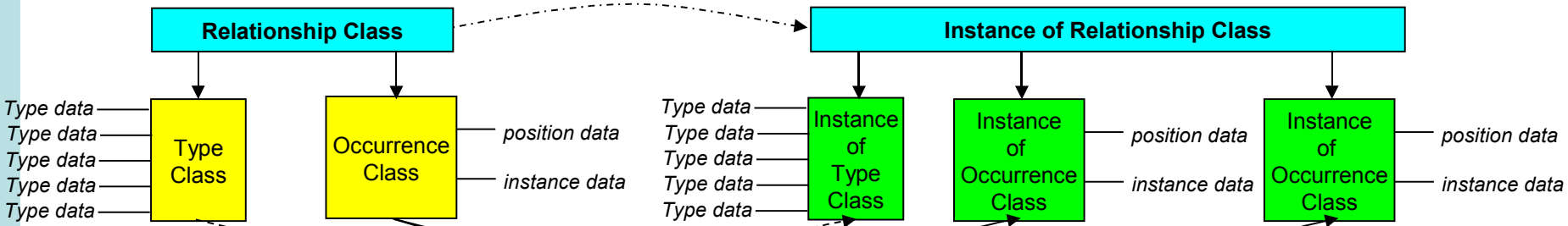
# Element Types

- Element types are specifications of the ‘real things’ in the IFC model
- Main concepts are
  - Building element types
  - Distribution element types
  - Furnishing element types
  - Equipment element types
- Tranport element types can be used
- Electrical element type now deprecated
  - improved electrical descriptions are now in distribution element types

# Type/Occurrence Classes

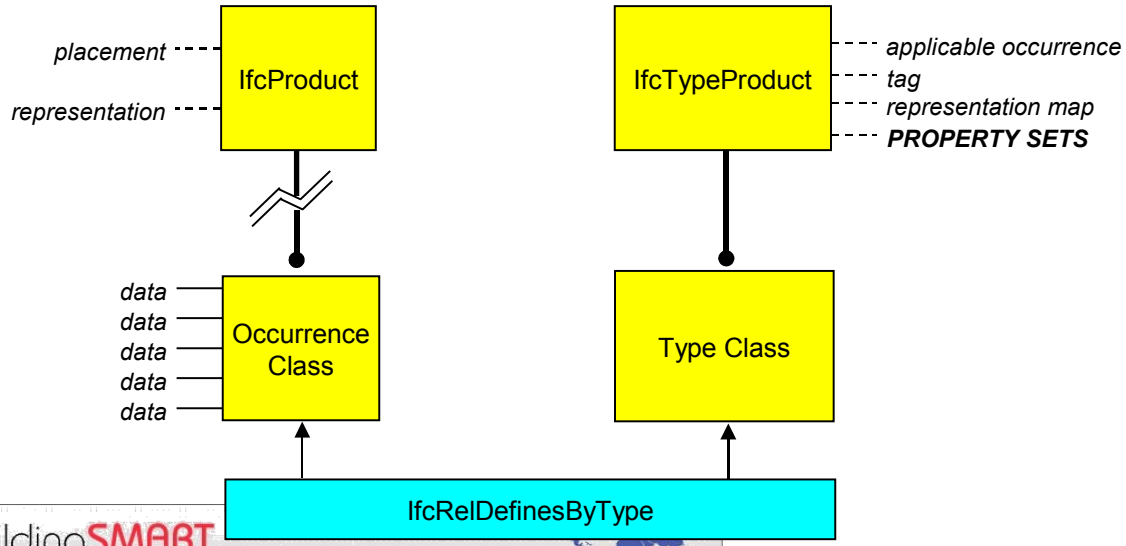
- A type class specifies the information to be captured about all instances of that type
- An occurrence class specifies placement (position) data and references the type.
- An occurrence class also specifies any instance specific data



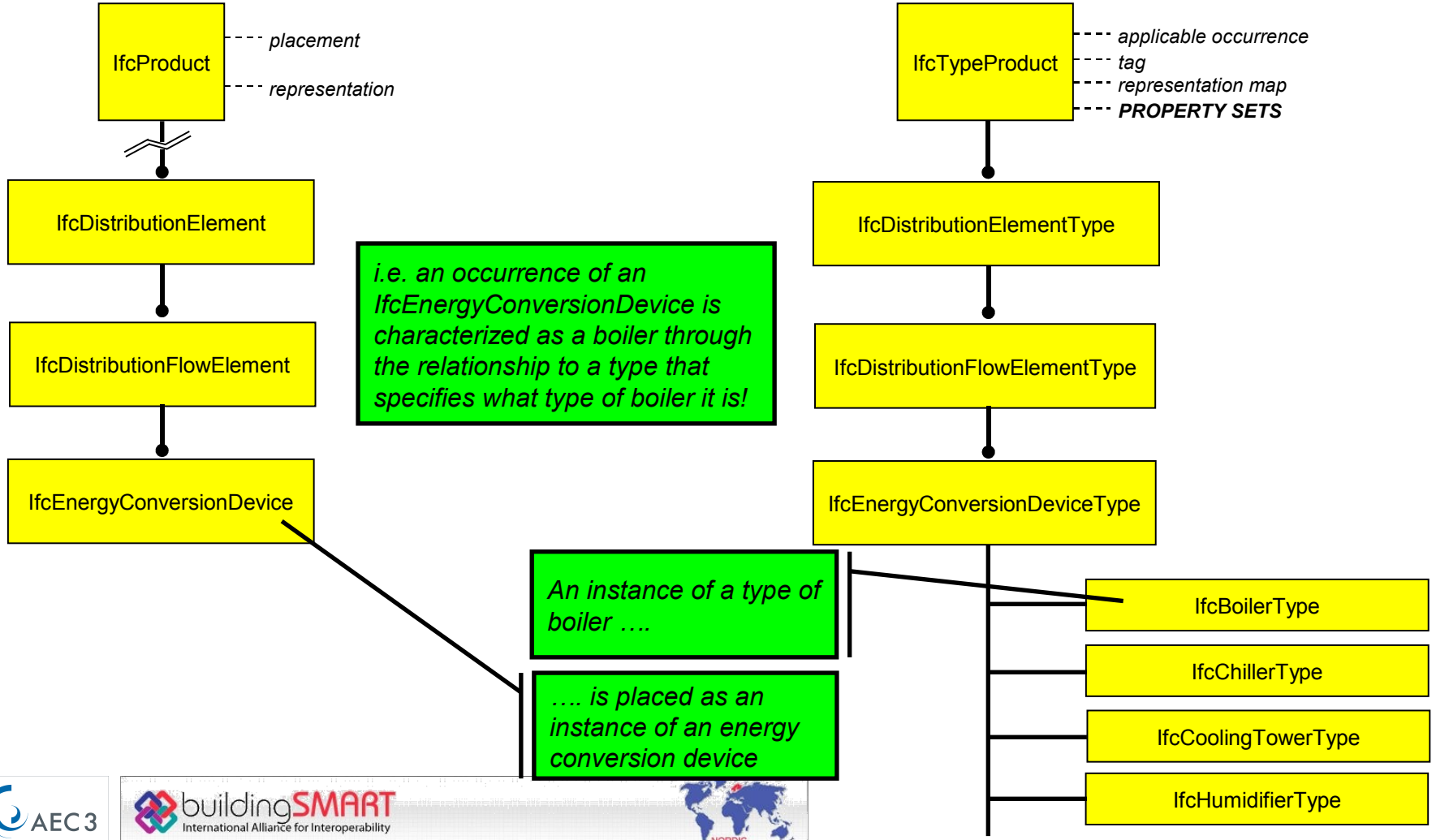
# Type/Occurrence Inheritance

- Occurrence classes are subtypes of IfcProduct allowing for placement and, if required, representation
- Type classes are subtypes of IfcTypeProduct
  - Can contain multiple property sets
  - Can use predefined shape representations (representation maps)
  - Data for types is normally given in property sets and not by direct attributes of the class



# Type/Occurrence Instances

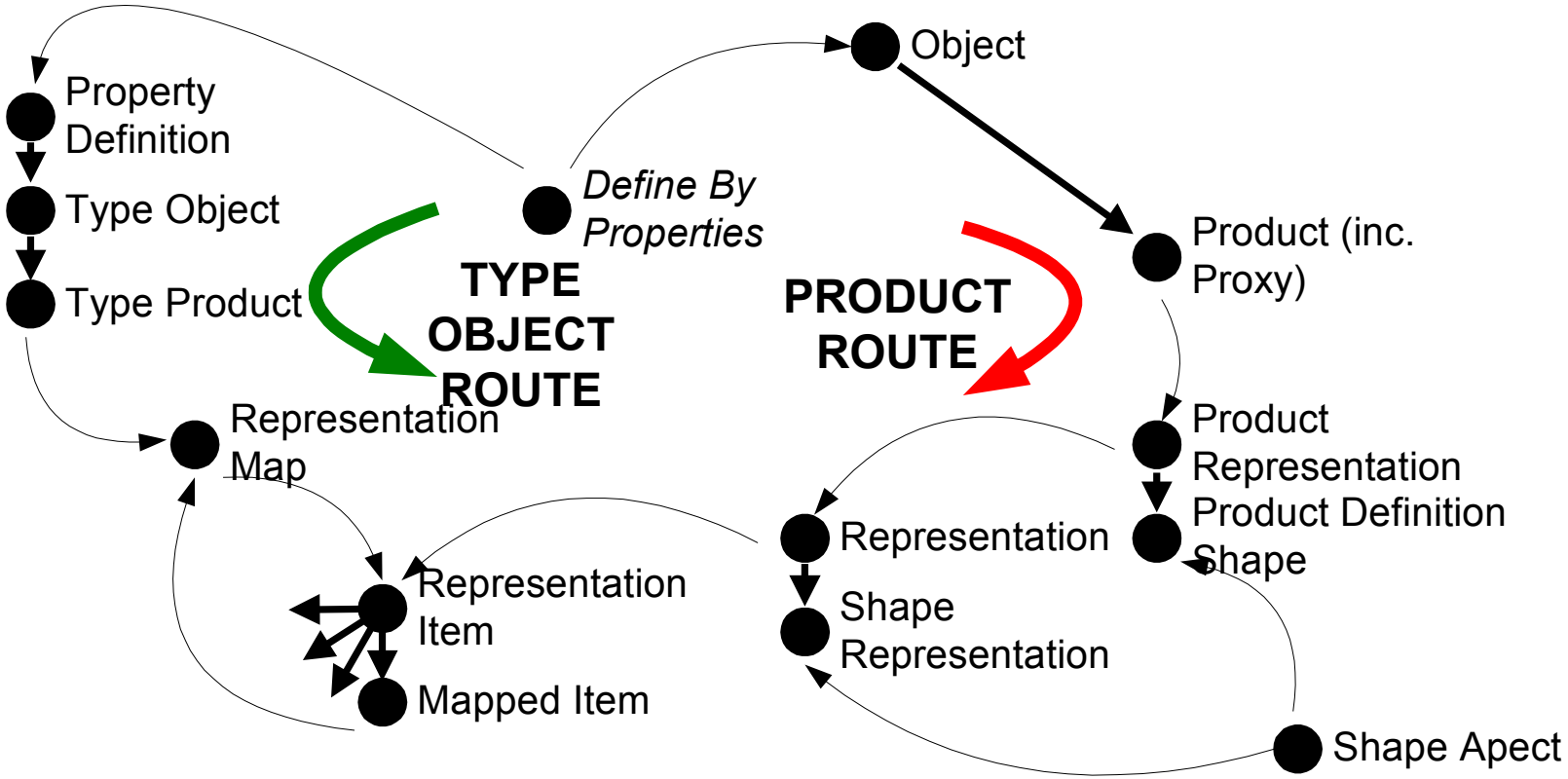
- The Type decomposition tree goes one level deeper than the Occurrence decomposition tree



# Geometry

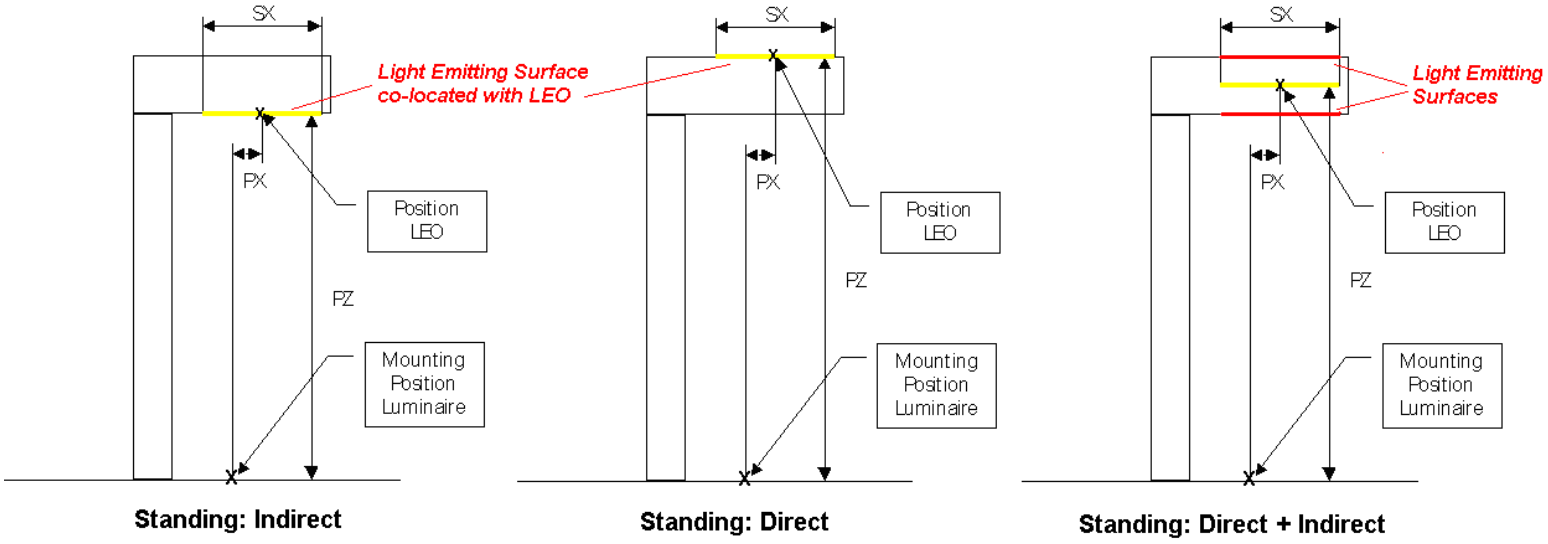
- This presentation will not go into details about IFC geometry
- BUT…
  - IFC geometry is an adapted version of ISO 10303 part 42 (STEP)
  - Robust, well proven, well used
  - Has a full range of geometry classes including for solids (CSG, B-rep), surfaces, curves, boolean operators, profiles
  - Has a full range of topology classes including shell, face, loop, edge, vertex, point, path
  - Can combine geometry/topology into representation maps (block/symbol/cell)
  - Can individually reference different aspects of shape within a representation (e.g. for sill of a door/window)

# Geometry Routes



# Presentation

- Presentation features in IFC include for
  - Draughting capabilities from ISO 10303 part 202
    - Linetype, lineweight, font type, font size, hatch, marker, layer, colour
    - Viewing pipeline for drawing layout not yet supported
  - Photometrically accurate rendering
    - to support VRML at a simple level
    - Lightworks, Radiance, 3DS at a more complex level



# Who can do IFC?

- All major BIM applications (geometric)
  - Autodesk ADT
  - Autodesk Revit
  - Bentley Architecture
  - Informatix microGDS
  - Graphisoft Archicad
  - Nemetschek Allplan
- Structural
  - Arche, Advance, Effel (Graitec)
  - BDP (Inféo)
  - Bocad
  - CSI
  - Dlubal
  - Friedrich & Lochner
  - Tekla
  - RIB
  - Robobat
  - Sofistik
  - Strusoft
  - InterCAD
  - RIB
  - …

**IFC-CIS2 Map**

- Building Services
  - DDS
  - Granlund
  - Lawrence Berkeley
  - A-NULL
  - Climawin
  - elcoCAD
  - RoCAD
  - Klima2000
  - CONTAMW
  - …
- Viewers
  - NavisWorks
  - Octaga
  - DDS
  - TNO
  - Univ. Karlsruhe
  - NavisWorks
  - …
- Database
  - EPM Technology
  - Oracle CBIM
  - Secom
  - Eurostep
  - STEP Tools

- Dictionary
  - STABU
  - Byggforsk
  - CSI
- FM
  - Vizelia
  - COBIE
  - …
- Costing
  - Focus
  - Tocoman
  - Timberline
  - Cadquant
  - Sumitomo
  - Kajima
- Building Codes
  - Fornax
  - PNNL
  - Solibri
- Early Design
  - Facility Composer
  - dRofus