buildingSMART International Policy statement

The relation between IFC and bSDD

Date: February 2020

Status: Final

Version: 1.0

Contents

Purpose of bSDD 3
Levels of connectivity 4
Publication and accessibility of data 4
Domain control 4
Changes needed in bSDD 4
Maintenance of IFC and bSDD 5
    Transporting bSDD data 5

Purpose of bSDD

Standardization of data is done on multiple levels. The lowest level is the level where everyone agrees on. The more specific things get, the more differences there are. This does not have to be a problem for interoperability, as long as the specific objects have a relation to the lower levels. IFC has a specialization structure, and every entity in a specific domain in IFC therefore has a shared element on a more generic layer.
At some point, it is impossible to standardize entities. This is where IFC has to stop. More local standards, or data standards for very specific use-cases (like 4D scheduling or 5D cost estimation) should not be part of IFC.
Definition and maintenance of these standards can be local. To maintain the interoperability with the 'foundation classes' it is important to have a relation to the representing IFC entity.

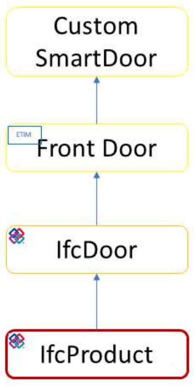

Figure 1: The specialization tree from generic IFC (IfcProduct) to a very custom specific door.

The definition of a '*wooden exterior front door'* is different per country, and the properties attached to it might be different depending on the use-case.
It is of great value, to define that this special kind of door is still a special representation of an IfcDoor. In some cases, users may have the need to define project-specific, or company-specific set of specializations and property sets. To facilitate the connections between these classification systems and property set definitions, the bSDD is created.

bSDD can hold data that extends the specialization tree of the Industry Foundation Classes. So bSDD classifications are extensions of the Industry Foundation Classes. There is also a need to extend IFC entities with additional properties. bSDD is also facilitating additional property definitions to classifications (IFC and other classifications).

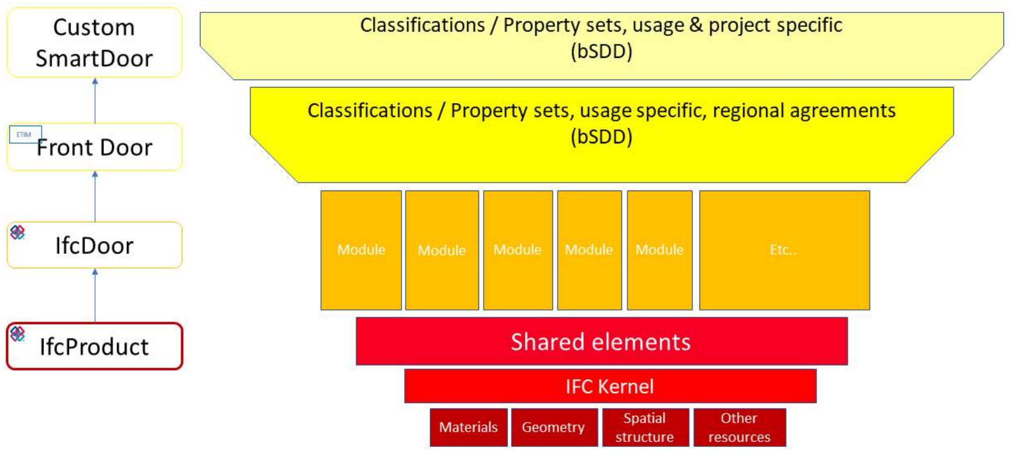

Figure 2: The specialization tree, with the role of the IFC resources, IFC interoperability layer (red), IFC Extensions (dark yellow) and the bSDD domains (yellow)

Levels of connectivity

Data inside the bSDD will have different levels of ‘connectivity’. Classification can just be in there without any connections, connections to IFC entities, or even to other entities in other domains.

1. None (only to data within the same domain)
2. To IFC (quality control needs to be in place, respecting the semantic definitions of IFC)
3. To each other (this is only possible when semantic descriptions are the same, alike, or something else. Type of linking needs to be elaborated depending on ISO12006 update and Linked Data ontologies).

Publication and accessibility of data

The data inside bSDD will be stored according to a suitable (canonical) data model. It will be accessible in multiple ways:

- As JSON web interface
- As static data according to (the updated) ISO12006-3
- As Linked Data

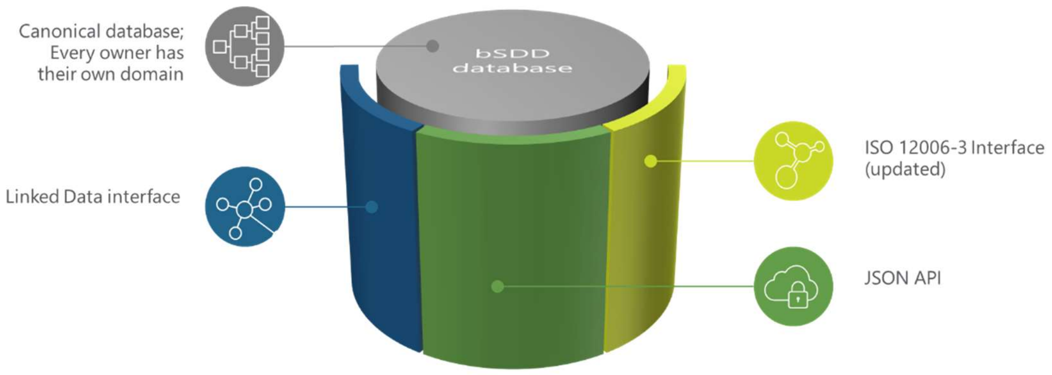
Canonical database; Every owner has their own domain
bSDD database
Linked Data interface
ISO 12006-3 Interface (updated)
JSON API

Figure 3: The three different ways to publish/access the data inside the bSDD database

Anonymous access will be available for a limited set of data. User registration should be available. Logged in users will have access to data depending on their access level.

Domain control

Only owners of data (or their agents) have the ability to update their data and suggest links to other domains. They also have the ability to set their data public, or under a license. This license has to comply with the overall publication license of the bSDD. Owners of data may charge users a fee for the use of their data (although this is a long-term feature of bSDD).

Changes needed in bSDD

Although classifications in bSDD can exist without a connection to other domains, it should be stimulated to embed them in the specialization tree of IfcProduct, IfcControl, or IfcProcess. In practice most of the entities will be part of the IfcProduct tree. When possible, extensions will be linked to the most specializes entity as possible. In the example mentioned before, the ‘Front Door’ is a specialization of an IfcDoor, and nót of an IfcProxy. The same goes for IFC PSets and properties.
The means IFC entities are the backbone of the bSDD, instead of one of the translations of the subjects. The current ‘subjects’ will be removed. As a consequence, the IFC Entities, Types and Enumerations need to be available in bSDD. Owner of this data is buildingSMART, and only buildingSMART will have the ability to maintain this data.

Maintenance of IFC and bSDD

The pyramid structure in Figure 2 shows the difference between IFC and bSDD. When there is an international agreement about entities, it will be in the red layer of the shared elements. International agreed domain specific entities and properties will be in the extensions (the modules). National and other agreements will be in bSDD. Keeping international (domain specific) agreements in IFC will facilitate automatic mapping and data analyses.

The core place to maintain IFC will be stored in UML/XMI. After every update, an EXPRESS, XSD and OWL dataset is being automatically generated using Continues Integration (CI) Tools from GitHub.

Because the official properties and Psets should be available in the bSDD as well, this infrastructure should be extended to also publish the Properties, Psets and PredefinedTypes in bSDD.

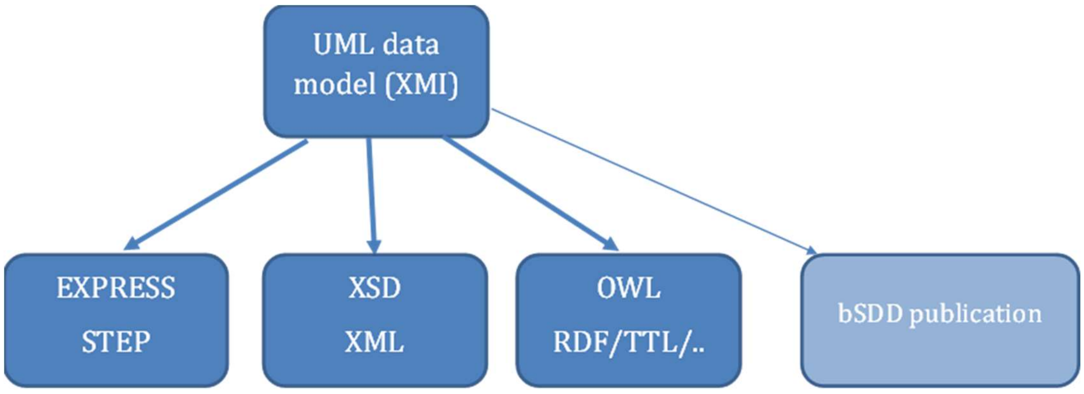

Figure 4: Generating and publishing the IFC entities and properties into the bSDD

Transporting bSDD data

The export of bSDD instantiated data (in case of an authoring tool) will most likely be in IFC. The object will have the base IFC entity class, with an additional classification reference and properties from the bSDD.
In the example above, the object will be exported as an IfcDoor, with a *IfcClassificationReference* 'Front Door'. The *IfcClassification* will reference to the URI of the bSDD object[1]. Alternatively, the use of *IfcLibraryReference* could also be considered.

It should be possible to use different library references in one IFC dataset (file).

[1] This might also need an update in IFC