buildingSMART action item

Resolution from buildingSMART International Council, Boston, 20-May-11 - IC 28/17

Notes with approval the plan to log the changes in the core between IFC2x3 and IFC2x4 (IFC4), with a view to discussing the consequences with interested users, including software developers, and ensuring that a correct balance is achieved between the benefits of any improvements and any adverse consequences to the users. The aim is to reach a conclusion by consensus between IUG and ITM. The proposal should be available for consultation by 19 July and for final resolution in Singapore.

Scope of document

This document contributes the MSG part to the ITM position on the compatibility issue, it solely focuses on the IFC schema relevant part regarding compatibility. It needs to be complemented by the software specific part, the buildingSMART communication and roll-out part and the user expectation part.

NOTE The document "Compatibility assessments for IFC4", written by TechCom members Thomas Liebich, Bjorn Stangeland, Jan Karlshoj and Rasso Steinmann provides a good overview about the full picture of compatibility.

The document is based on an in-depth comparison between IFC4 and IFC2x3 schemas, a test and evaluation of how different IFC development toolboxes react on changes, and a summary and explanation of the findings.

Issue description

Concerns have been raised about the compatibility between the currently implemented and supported IFC release IFC2x3 and the upcoming new release IFC4.

NOTE The general issue description is already explained in detail within the buildingSMART TechCom paper "Compatibility assessments for IFC4". Please refer to this document.

Any enhancement that leads to a new schema release of IFC will generate a compatibility issue. At this point it is independent of the magnitude of the change, as any new schema release results in a new schema name.

```
IFC2x3 header contains
    FILE_SCHEMA(('IFC2X3'));
IFC4 header contains
    FILE_SCHEMA(('IFC4'));
```

This change in schema name would lead to an error message, such as "unsupported schema, stopped import", if there is no general awareness by the software system.

ADVISE The IFC compatibility issue cannot be seen as an isolated IFC schema issue. It needs to be addressed by a combination of schema design, software support and user awareness – a rollout strategy is needed.
IFC schema compatibility assessment

From now on this document concentrates solely on the issue of schema compatibility¹. The following definition of schema compatibility is used (based on the broad agreement in product modelling community, coming from the ISO STEP developments):

Schemas are deemed to be backward compatible, if the physical data exchange structure remains unchanged for all entities defined in the previous release.

To illustrate:

In IFC2x3 the inverse relationships between an object and its properties, and an object and its type are the same. This is semantically weak and leads to ineffective implementations (the need to search through all properties to find the type). In IFC4 there is a dedicated inverse relationship to identify the type directly. Since inverse attributes are not part of the physical exchange, such changes are deemed to be fully backward compatible. Software implementations however have to adapt to the new schema.

```
IFC2x3 physical exchange structure:

#100= IFCBEAMTYPE('0juf4qyggSstrxA20Qwnsj',$,'IPE220','Beam type',$,$,$,$,$,.BEAM.);
#1000= IFCBEAM('0juf4qyggSI8rxA20Qwnsj',$,'A-1','IPE220','Beam',#1001,#1010,'A-1');
#90010=IFCRELDEFINESBYTYPE('2aq$Crcs_xJvd69lbm2bMM',$,'beam typing',$,(#1000),#100);
```

```
IFC4 physical exchange structure

#100= IFCBEAMTYPE('0juf4qyggSstrxA20Qwnsj',$,'IPE220','Beam type',$,$,$,$,$,.BEAM.);
#1000= IFCBEAM('0juf4qyggSI8rxA20Qwnsj',$,'A-1','IPE220','Beam',#1001,#1010,'A-1',$);
#90010=IFCRELDEFINESBYTYPE('2aq$Crcs_xJvd69lbm2bMM',$,'beam typing',$,(#1000),#100);
```

NOTE This example shows that no change of the exchange structure is reflected in the IFC data files, hence it is deemed to be backward compatible.

When defining IFC4 as the major next IFC release an enhanced backward compatibility definition has been used. In addition to the above, also those changes are deemed to be backward compatible that lead to changes in the physical exchange structure that can be resolved by available toolboxes with no data losses.

IFC schemas are deemed to be backward compatible, if the physical data exchange structure only contains changes for all entities defined in the previous release that can be resolved without data loss.

(and with available toolboxes and without additional mapping functions)

To illustrate:

In IFC4 the IFC schema has been unified to have the same object occurrence and type breakdown and use of predefined types throughout the entire schema. This allows software applications to use generic code and will lead to more reliable exchanges. As a consequence the IFC4 entity Beam for example has an additional attribute "predefined type", that did not exists in IFC2x3. When reading into an IFC4 toolbox, the missing attribute will be added with undefined value. As such value did not exists in IFC2x3, an undefined value represents no data loss.

¹ see the document "" for all definitions, such as "backward compatibility", "forward compatibility", etc.
**IFC2x3 physical exchange structure:**

```
#1000= IFCBEAM ('0juf4qyggSI8rxA20Qwnsj','$,
'A-1','IPE220','Beam',#1001,#1010,'A-1');
```

**IFC4 physical exchange structure**

```
#1000= IFCBEAM ('0juf4qyggSI8rxA20Qwnsj','$,
'A-1','IPE220','Beam',#1001,#1010,'A-1',$);
```

NOTE This example shows that the last attribute got added with an undefined value (represented by "$"). This represents a change with no data loss.

**Detailed change assessment**

Based on the decision by the buildingSMART ITM committee the whole scope of the only currently buildingSMART certified IFC2x3 model view definition, the IFC2x3 Coordination View V2.0, has been surveyed in order to summarize all changes and the reason for the change.

**ITM resolution #47-1**

Compare the IFC2x3 Coordination View 2.0 with its counterparts in IFC4 CV to document the actual changes in IFC4 with reasons for necessity.

The result is shown in Appendix II. It represents an in-depth analysis of all changes (and areas on no change), the reason for the change and the compatibility assessment for each change.

The other buildingSMART ITM committee resolution has been:

**ITM resolution #47-2**

Prepare a second comparison of a major (already implemented) MVD, i.e. the CDB 2010, in the comparison. This will require that CDB 2010 be documented as a legal Express subset of the Schema of IFC2x3. This is not like to be completed by mid-May due to the latter requirement.

This comparison has not been done, since the CDB2010 model view definition has not been made available as a legal Express subset of the IFC2x3 schema.

The final buildingSMART ITM committee resolution regarding the compatibility issue has been:

**ITM resolution #47-3**

Select the most important changes and additions made in IFC4 (compared to IFC2x3) and explain in end-user terms what capabilities and benefits it would provide when implemented.

The detailed result is shown in Appendix I, it contains an in-depth assessment of each change pattern and its effect on compatibility. In addition we have also made a survey of available IFC toolboxes to test how those changes are handled and whether the backward compatibility claim is maintained. In regard to the benefits in user terms, please refer to Appendix II column "reason for change" that describe what can be expected.

**Assessment of IFC4 schema change patterns**

The following table shows the summary of all change patterns found when analysing the changes found in Appendix I.

<table>
<thead>
<tr>
<th>Page no.</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Thomas Liebich</td>
</tr>
</tbody>
</table>

NOTE Those changes are already indicated in the IFC2x4 intermediate schema publications that are published since two years. None of those are new, or had been hidden in the past.
<table>
<thead>
<tr>
<th>Type of change</th>
<th>Compatibility assessment</th>
<th>IFC file changes</th>
<th>Information loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enumerators added</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Enumerators deleted</td>
<td>partially backward compatible with warning</td>
<td>unknown enum value</td>
<td>attribute value might not be available</td>
</tr>
<tr>
<td>Enumerator renamed</td>
<td>partially backward compatible with warning</td>
<td>unknown enum value</td>
<td>attribute value might not be available</td>
</tr>
<tr>
<td>Inverse attribute renamed</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Inverse attribute added</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Inverse attribute cardinality changed</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Inverse attribute type changed</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Inverse attribute deleted</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Attribute renamed</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Attribute type changed (more general)</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Attribute made mandatory (was optional)</td>
<td>backward compatible with warning</td>
<td>unexpected &quot;$&quot; value</td>
<td>no information loss (attribute was and is undetermined).</td>
</tr>
<tr>
<td>Attribute made optional (was mandatory)</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Attribute added (at end and optional)</td>
<td>backward compatible with warning</td>
<td>missing attribute(s) at end of entity (too few)</td>
<td>no information loss (non-existing attribute set to undetermined)</td>
</tr>
<tr>
<td>Attribute added (not at the end)</td>
<td>not backward compatible</td>
<td>reordering of attribute sequence in entity</td>
<td>information loss of whole entity information - instability.</td>
</tr>
<tr>
<td>Attribute deleted (last attribute)</td>
<td>backward compatible with warning</td>
<td>additional attribute(s) at end of entity (too many)</td>
<td>none - additional old attribute lost (but it do not exist in new schema)</td>
</tr>
<tr>
<td>Attribute deleted (not last attribute)</td>
<td>not backward compatible</td>
<td>reordering of attribute sequence in entity</td>
<td>information loss of whole entity information - instability.</td>
</tr>
<tr>
<td>Attribute demoted to subtype (no change of order)</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Attribute promoted to supertype (no change of order)</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Type of change</td>
<td>Compatibility assessment</td>
<td>IFC file changes</td>
<td>Information loss</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Entity added (new subtype)</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Entity deprecated</td>
<td>fully backward compatible</td>
<td>(unknown entity name – but announced well ahead) none</td>
<td>(based on software acknowledging deprecations) none</td>
</tr>
<tr>
<td>Entity deleted</td>
<td>not backward compatible</td>
<td>unknown entity name</td>
<td>information loss of whole entity information</td>
</tr>
<tr>
<td>Entity deleted (was deprecated)</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Abstract entity made non-abstract</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Abstract entity renamed</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Entity made abstract</td>
<td>not backward compatible</td>
<td>entity name marked as ABSTRACT in schema</td>
<td>potential information loss of whole entity information</td>
</tr>
<tr>
<td>Supertype changed without reordering attributes</td>
<td>fully backward compatible</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Single attribute changes to list (set, bag) type</td>
<td>partially backward compatible with warning</td>
<td>simple attribute, where set '{' is expected</td>
<td>attribute value might not be available</td>
</tr>
<tr>
<td>List (set, bag) attribute changes to single attribute</td>
<td>partially backward compatible with warning</td>
<td>set '{' where simple attribute is accepted,</td>
<td>attribute value might not be available</td>
</tr>
</tbody>
</table>

Table 1: detailed assessment of IFC4 schema change pattern

As a result MSG will fix all changes shown in red with the upcoming IFC2x4 RC3 release. Those changes shown in orange need to be further evaluated based on a comparison of the value of change versus the issue of compatibility. All other changes shown in green or yellow should remain as they provide added value (new content, better schema consistency, etc.) with no or easy to achieve backward compatibility on schema level.
Conclusions

The following conclusions are made:

- the compatibility issue needs to be addressed by schema design, software implementation and guided user expectations,
- the schema design of IFC4 has been assessed and only few incompatible changes regarding backward compatibility are detected, those will be fixed,
- the vast majority of changes are deemed to be backward compatible regarding the schema
- no necessity to make any drastic changes to IFC4 (based on the current version IFC2x4 RC2) could be found – the recommendation is to continue the IFC4 development and its recognition as a full ISO standard ISO 16739 while addressing the few non backward compatible changes one by one.

under the following assumptions:

- schema compatibility only addresses backward compatibility, forward compatibility (and backward compatibility of non-core parts) need to be achieved using mapping services
- software applications need to be aware of backward compatibility (including accepting previous schema name, surprising "no information loss" change warnings, updating the API to the new schema for changed queries, etc.)
- user expectations need to be clear – there is no improvement without issues regarding backward compatible changes – it is always a balancing of what has to be new, and what has to be unchanged
- buildingSMART needs to define a rollout strategy when publishing IFC4. The ISO acceptance process for ISO 16739 also sets requirements that need to be obtained.

It should be noted that the assessment had been made based on the only buildingSMART certified model view definition of IFC2x3 that included many parts that are not part of the stable IFC2x core. So many of these compatibility requests go far beyond the original IFC2x core stability claim and cannot be demanded by referring to it.