



**International Alliance for Interoperability
Modeling Support Group**

ifcXML

Implementation Guide

Editors: Nick Nisbet, Thomas Liebich

**Version 2.0
June 29th, 2007**



All rights reserved. No part of the contents of this document may be reproduced or transmitted in any form or by any means without the written permission of the copyright holder (IAI).

Copyright © 1996-2007 - International Alliance for Interoperability (IAI)

Document Control

Editor	Nick Nisbet, Thomas Liebich
Contributors	P. Houbaux, G Skarseth, Francois Grobler
Development Committee	Model Support Group
Project Reference	ifcXML implementation support
Document Reference	ifcXML Implementation Guide
Document Version	Version 2.0
Release Date	June 29, 2007
Status	
Distribution	Public
Distribution format	PDF file

Table of Content

DOCUMENT CONTROL	2
DOCUMENT HISTORY	6
1 INTRODUCTION	7
1.1 PREFACE	7
1.2 ABOUT THE EDITORS	7
1.2.1 <i>Nicholas Nisbet MA(Cantab) DipArch(UNL)</i>	7
1.2.2 <i>Dr.-Ing. Thomas Liebich</i>	7
1.3 PURPOSE	7
1.4 TARGET AUDIENCE.....	8
1.5 BUSINESS MOTIVATION	8
1.6 EXAMPLE APPLICATION AREAS.....	8
1.7 EXAMPLE APPLICATIONS	8
1.8 COMPARATIVE ADVANTAGES.....	9
1.9 OTHER REFERENCES	10
1.10 DOCUMENT CONVENTIONS	11
2 IFCXML GENERAL DISCUSSION	12
2.1 STATUS	12
2.2 DEFINITION	12
2.2.1 <i>XML and XLINK</i>	13
2.2.2 <i>XSD</i>	13
2.3 XML DOCUMENTS	17
2.3.1 <i>Entities</i>	18
2.3.2 <i>Namespace</i>	18
2.3.3 <i>Properties</i>	23
2.4 LIMITATIONS	24
2.4.1 <i>Known limitations</i>	24
2.4.2 <i>Possible application limitations</i>	24
2.5 CURRENT IMPLEMENTATIONS OF IFCXML	25
2.5.1 <i>EPM Express Data Manager v4.7 to v5.0</i>	25
2.5.2 <i>Eurostep Model Server (EMS)</i>	25
2.5.3 <i>XMLSPY 2007</i>	26
2.5.4 <i>TurboXML</i>	26
2.5.5 <i>Octaga Modeller</i>	26
2.5.6 <i>ArchiCAD v9.0 to v11.0</i>	26
2.5.7 <i>AEC3 Ltd</i>	26
2.5.8 <i>Research Center of Karlsruhe, Germany</i>	26
2.5.9 <i>TNO, Netherlands</i>	26
2.5.10 <i>ifcSVR300</i>	26
2.5.11 <i>ALLPLAN2006.2</i>	26
2.5.12 <i>ECCO (EXPRESS toolbox by PD Tec)</i>	27
3 USE CASES 28	
3.1 USE CASE: MATERIAL CATALOGUE	29
3.1.1 <i>Description</i>	29
3.1.2 <i>Purpose:</i>	29
3.1.3 <i>Actors:</i>	29
3.1.4 <i>Transactions:</i>	29
3.1.5 <i>Applications:</i>	29
3.1.6 <i>Data content</i>	29
3.1.7 <i>Comment</i>	30
3.2 USE CASE: BILL OF QUANTITIES REPORT	31
3.2.1 <i>Description</i>	31
3.2.2 <i>Purpose:</i>	31
3.2.3 <i>Actors:</i>	31

3.2.4	Transactions:	31
3.2.5	Applications:	31
3.2.6	Data content	31
3.2.7	Comment	33
3.3	USE CASE: ADDING USER DEFINED PROPERTIES	34
3.3.1	Description	34
3.3.2	Purpose:	34
3.3.3	Actors:	34
3.3.4	Transactions:	34
3.3.5	Applications:	34
3.3.6	Data content	34
3.3.7	Comment	35
4	WRITING IFCXML	36
4.1	MANDATORY CONTENT	36
4.1.1	Mandatory header and footer	36
4.1.2	Mandatory elements	36
4.1.3	Mandatory attributes	36
4.2	EXAMPLE MINIMAL (EMPTY) FILE	37
4.3	TYPES	39
4.3.1	Type matching using href and ref	39
4.3.2	Bag, Set or List	39
4.4	ORDER	39
4.4.1	Alphabetical	39
4.4.2	Dependency	39
4.4.3	Reversed dependency	39
4.5	PRIORITIES	39
4.5.1	Size	39
4.5.2	Simplicity	40
4.5.3	Speed	41
4.6	VALIDATION	41
4.6.1	XML	41
4.6.2	ifcXML	41
4.6.3	IFC	42
5	READING IFCXML	43
5.1	STARTING	43
5.2	ALTERNATION	43
5.3	HREF LINK	43
5.4	NAMESPACE	43
5.5	ESTABLISHING UNITS	43
5.6	FILTERING	43
6	CODE SAMPLES	44
6.1	XSLT	44
6.2	</XSL:STYLESHEET>JAVA	45
6.3	C++	45
6.4	VB	45
7	QUESTIONS AND ANSWERS	46
7.1	WHY SHOULD I USE IFCXML?	46
7.2	DO I HAVE TO LEARN AND UNDERSTAND THE WHOLE IFC MODEL ?	46
7.3	WHAT IS THE BEST WAY TO REALLY UNDERSTAND THE IFC MODEL?	46
7.4	WHAT ARE IFC MODEL VIEWS AND HOW DO I DEAL WITH VIEWS?	46
7.5	DOES IFCXML REPLACE SPF FORMAT?	46
7.6	DO I HAVE TO LEARN EXPRESS AND SPF FORMAT TO USE IFCXML?	46
7.7	WHY IS IFCXML MORE COMPLICATED THAN TEXT BOOK EXAMPLES OF XML?	47
7.8	CAN I SEND OR RECEIVE PARTS OF A WHOLE BUILDING MODEL?	47
7.9	DO I NEED AN OBJECT MODEL SERVER TO WORK WITH IFCXML MODELS?	47
7.10	CAN I STORE AND RETRIEVE IFCXML DATA USING MS ACCESS DATABASE?	47

7.11	HOW DOES IFCXML RELATE TO AECXML?	47
7.12	HOW DOES IFCXML RELATE TO GBXML?	47
APPENDIX 1	EARLIER IFCXML METHODOLOGIES	48
A 1.1	HISTORY	48
A 1.2	DIFFERENCES	48
A 1.3	EXAMPLE (SUPERCEDED)	48
A 1.4	LIMITATIONS	49
A 1.5	IMPLEMENTATIONS (UP UNTIL JULY 2004)	50

Table of Figures

FIGURE 1: THE ANTICIPATED ROLE OF SPF AND IFCXML	9
FIGURE 2: IFCXML AND ALTERNATIVES	10

Document history

Document version 2.0, Author N. Nisbet, T. Liebich date 29.06.2007

- New ifc2x3 release
- Final ifc2x2 release
- New URIs for ifcXML2x2 namespace
- Revisions and corrections for name space and prefix added in 2.3.2

Document version 1.0, Author T. Liebich, date 10.02.2005

- Preparation for issuing publicly
- Reference to ifcXML update (ifcXML-2 for IFC2x2 Addendum1)
- examples for name space and prefix added in 2.3.2

Document version 0.7, Author N. Nisbet, date 25.09.2004

- Minor corrections
- Octaga Viewer updated
- Header details in 4.1.1 corrected

Document version 0.6, Author N. Nisbet, date 20.09.2004

- Comments from P Houbaux, particularly on partial models working with a server.

Document version 0.5, Author N. Nisbet, date 09.09.2004

- First release
- Punctuation corrections.
- Figures 1 and 2 added.

Document version 0.4, Author N. Nisbet, T. Liebich date 02.09.2004

- FAQ section added with material from Francois Grobler
- Property set example added

Document version 0.3, Author N. Nisbet, T. Liebich, date 09.08.2004

- RDMS limitations added
- Two use cases indicated
- More explanations for the definition section (see section 2.2).
- More explanations for Part28 edition 2 (see section 2.2.2.1).
- Incorporated comments from P. Houbeaux
- Moved Status to section 2
- Examples added
- More examples added for section 2.3.
- Document conventions added (see section 1.7) and enforced
- OwnerHistory requirement added (see section 4.1.3)
- Bill of Quantities example added (see section 3.2)

Document version 0.2, Author N. Nisbet, T. Liebich, date 06.08.2004

- XSLT example added
- All references to ifcXML-1 methodology removed to Appendix
- Formal title page added
- Paragraph numbering corrected

Document version 0.1, Author T. Liebich, date 05.08.2004

- Formatting of IFC keywords
- Addition to the definition of terms ifcXML methodology, schema, file
- Several discussion items included

Document version 0, Author N. Nisbet, date 12.07.2004

- Initial set-up of document
- For private discussion

1 Introduction

This document targets the XML community by providing guidelines on using and implementing the IFC standard using XML technologies. It is an extension to the existing IFC literature with focus on the specifics of the ifcXML specification compared to the standard EXPRESS based IFC model (as documented in the IFC object model specification and the IFC implementation guide).

1.1 Preface

This guide was first produced in the Summer of 2004 in response to increasing worldwide interest in ifcXML. It is based on the several years experience of defining and implementing the emerging standard and correspondence with experts and students. The second edition updates the material and includes references to ifc2x2 and also ifc2x3.

1.2 About the editors

The authors welcome contributions and comments that will be considered for inclusion in later editions.

1.2.1 Nicholas Nisbet MA(Cantab) DipArch(UNL)

Nicholas Nisbet is an Architect who has been involved with many significant landmarks in the development of Computer Aided Architectural design and Building Information Modeling.

In 2002 he joined Jeffrey Wix Consulting, becoming a director a year later. He specializes in the implementation of IFC based applications and was the technology consultant to the UK Government sponsored two year project on IFC model based Operation and Maintenance of Buildings.

Nicholas Nisbet is active in the International Alliance for Interoperability (IAI) where he is an observer on the Model Support Group and Implementation Support Group and leading the ifcXML implementation project.

Email: nn@aec3.com

1.2.2 Dr.-Ing. Thomas Liebich

As a graduate in architecture Thomas Liebich continued with academic research into knowledge based systems, which he finished with the doctoral degree at the Bauhaus University in Weimar, Germany in 1993.

After working for private consulting companies, Thomas founded the company TLConsulting based in Munich in 1996, working on the fields of Information Technologies for the Construction Industry.

As the leader of the Modeling Support Group of the Industrial Alliance for Interoperability, IAI, he has overall responsibility for definition of the Industry Foundation Classes, IFC. He is the lead developer of the ifcXML specification.

Email: tl@aec3.com

1.3 Purpose

This ifcXML Implementation guide is intended to make accessible advice and guidance on using the XML representations of the IAI IFC schema and data.

ifcXML data files are normally given the extension .xml or occasionally .ifcxml or .ifx. The ifcXML representation is an alternative to the Step Physical File (SPF) representation. IFC files in the SPF files are normally given the extension .ifc or occasionally .stp.

It is anticipated that ifcXML will be of interest to a wider but less focused community than the implementers of the SPF which is currently the primary representation within the AEC-FM domain.

Note: The term ifcXML covers the methodology for generating the XSD out of the IFC EXPRESS definition, the XML schema (XSD) and also XML documents written against that ifcXML schema, see section 2.2 for more explanations.

1.4 Target audience

The target audience for this document is envisaged to include AEC-FM application developers, who may be familiar with the IFC schema, but also developers and implementers from other domains. Developers should have a good background knowledge of XML technology.

1.5 Business motivation

In offering an XML representation of IFC data, it is anticipated that a broader community of applications will be able to access a unified schema representing the built environment and related resources. XML has a broader range of supporting utilities and database implementations and is the basis for most eCommerce messages and Web services. XML is also supported by some web browsers especially with XSLT style sheets, making the information immediately accessible on workstations and most other handheld computing devices.

By adhering to the IFC content and semantic the XML community will benefit from an internationally accepted and agreed upon standard for the AEC-FM industry that has already been widely tested in several domains using other representations.

1.6 Example application areas

It is anticipated that ifcXML will have a role in application areas such as mapping between the IFC object model and:

- Document based representations such as schedules, quantity take-offs and product datasheets
- Message based representations such as Requests-For-Information (RFI)'s, orders and other eCommerce communications.
- Communication with XML based domains, such as the GIS object models based on the gml3 standards

The ifcXML representation is also expected to facilitate the extraction, transmission and merging of partial models during AEC-FM processes. Partial models are particularly significant in parallel and collaborative design processes.

1.7 Example applications

Applications supporting ifcXML are not envisioned to replace those supporting IFC SPF data. Rather, they are seen as complementary to existing applications. Potential applications that could benefit from the use of the ifcXML representation include:

Delivering to AEC-FM

- product and material libraries
- briefing
- maps and GIS context

Delivering from AEC-FM

- eCommerce messages
- asset registers
- maps and GIS updates

- BMS (building monitoring systems)
- visualization and VR
- cost applications.

In general it is anticipated that the exchange of complete building models, particularly within a CAD environment will continue to be primarily based on SPF format. The SPF format allows for more compact file sizes, while still remaining an ASCII format. Therefore exchange scenarios where large information sets (often including geometric models) are shared would remain to be the focus of the current SPF file format.

Other scenarios where partial models, reports, schedules or set of manufacturer information are exchanged, where file size is not such an issue, and where product databases and office applications are involved, etc. would benefit from the new XML format for IFC interoperability.

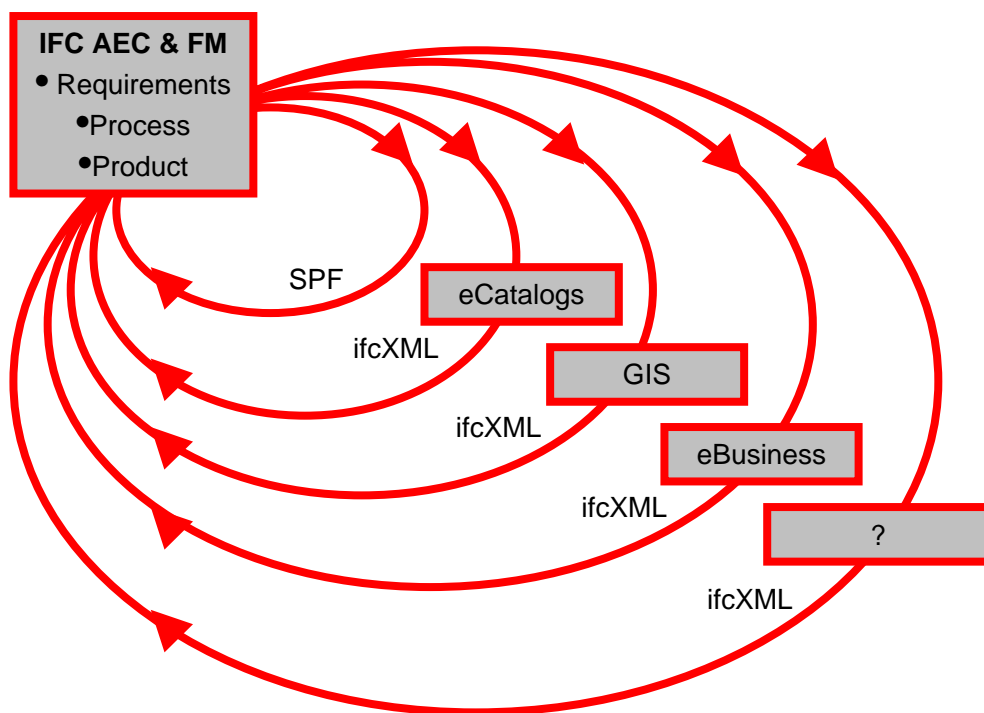


Figure 1: The anticipated role of SPF and ifcXML

1.8 Comparative advantages

There are four principal methods in working with IFC data

- Express STEP SPF format including various toolkits
- Express STEP SDAI (VB, Java and C++) bindings to a product model server
- Express-X and other database query and modification languages.
- ifcXML including various applications and interfaces

Compared with the other three methods, ifcXML is currently less common. As a file it is significantly larger than SPF, between 2 and 10 times larger, which may be significant if the content is a whole project model. However the main purpose of ifcXML is not seen to replace full project model exchange (where file sizes for SPF can easily exceed 50MB), but to cover partial exchange and reports, documents, queries, etc. where file sizes are only a fraction.

Some XML tools are less capable when working with very large documents. Where performance is critical, IFC models should be accessed from a model server or database. Some databases support the execution of EXPRESS or XSLT transforms directly on the data held.

The ifcXML schema is a derivation from the Express schema which is richer. The mapping to the XML schema definition (.xsd) necessarily loses some constraints including rules, inverse relationships and derived attributes. ifcXML data that validates against the ifcXML schema (.xsd) may not be fully conformant to the full IFC EXPRESS schema (.exp).

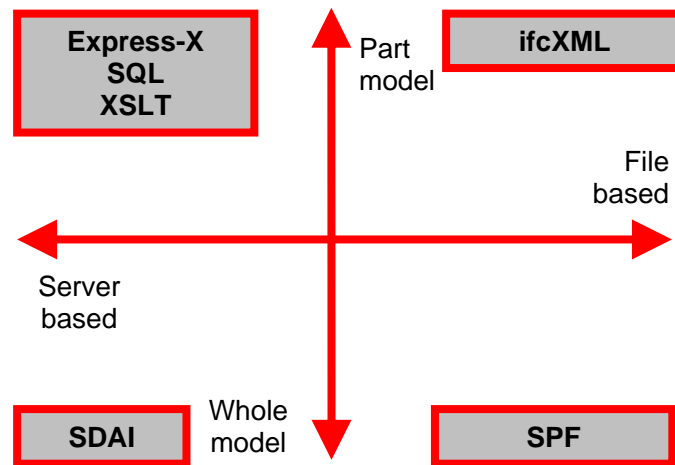


Figure 2: ifcXML and alternatives

1.9 Other references

This guide should be read in close conjunction with the IFC schema documentation and the IFC implementation guide. The examples given in the implementation guide are in the more concise but cryptic SPF format. The IAI IFC schema, the SPF format and ifcXML formats are a particular implementation of the ISO-10303 standards and reference can be made to these documents as the ultimate authority.

The official ifcXML specification is published on the IFC XML pages. It contains a reference to the ifcXML methodology (Part28 configuration file), and the ifcXML schema for IFC2x3 and IFC2x2.

- [http://www.iai-international.org/Model/IFC\(ifcXML\)Specs.html](http://www.iai-international.org/Model/IFC(ifcXML)Specs.html) (till 2006)
- <http://www.iai-tech.org/> (new location since 2007)

The documentation of the individual elements (what elements and attributes actually mean) are the same as in the IFC specification (EXPRESS based), they can be looked-up by searching the latest IFC specification. The official IFC EXPRESS specifications and the IFC model implementation guide can be accessed at:

- http://www.iai-international.org/iai_international/Technical_Documents
- http://www.iai-international.org/Model/R2x3_final/index.html

NOTE: The best entry point into the documentation is often the "alphabetical listing" under "Entities". Each parent XSD element has a corresponding EXPRESS entity that contains the documentation of itself, and its attributes (the child elements in XSD).

IFC implementation is also guided by Implementers Agreements which are monitored by the IAI Implementation Support Group (ISG), made up of active software developers, and assistance can be sought from them individually or collectively if necessary. The IAI and ISG also maintain online collaboration space (using <http://www.groove.net>) for its members to support and coordinate implementation. The home page of the ISG can be accessed at:

- <http://www.iai.fhm.edu/>

Development of the IAI IFC schema is guided by the Model Support Group and issues not handled by the ISG can be passed to the MSG for clarification.

Guidance is widely available on XML and XLINK implementation, based on the W3C definitions. This guide is subsidiary to these references, and contains comments which are the opinion of the authors and may not reflect the opinion of the IAI or its chapters.

Further links:

These general pages provide information about XML and XML schema:

- <http://www.w3.org/XML/Schema>
- <http://xml.coverpages.org/facilitiesXML.html>

1.10 Document conventions

Extracts from schemas and files are presented in Courier New

Hyperlinks are colored blue for example

<http://xml.coverpages.org/facilitiesXML.html>

IFC entities are colored blue for example `IfcSpaceBoundary`

IFC properties and attributes in running text are quoted for example 'Longitude'

IFC data is shown in bold and in red for example

```
<IfcMaterial><Name>Concrete</Name></IfcMaterial>
```

Examples of complete files follow the following convention:

- Green: Directives
- Red: Attribute names
- Brown: Entities
- Blue: Structured mark-up
- Black: Data

For compatibility with IFC standards, spelling is American English. Only elements taken directly from the ISO/STEP standard 10303 are spelled in British English.

2 ifcXML general discussion

2.1 Status

The ifcXML representation is an implementation of the ISO-10303 Part 28 Edition 2 (“part 28”) standard. This standard provides an XML schema specification that is an automatic conversion from the EXPRESS (ISO 10303 part 1) representation of the IFC schema. The mapping from EXPRESS to XML schema is guided by a configuration file that controls the specifics of the translation process. For ifcXML this configuration file is standardized and published for each version of the corresponding IFC schema.

The ifcXML2 methodology has now been approved for the generation of XML schema definitions for IFC2x Edition 2 and forthcoming IFC releases. The first schema, ifcXML for IFC2x2 was unveiled publicly on January 28th 2004, in Paris during the IAI summit. It currently has the release candidate 1 (RC1) status. After intensive testing it is anticipated that a final recommendation will be officially announced during the next IAI summit. A new ifcXML schema (using the ifcXML2 methodology) will also be created for the IFC2x2 Add1 release, also taking updates of part28 (that is currently in the ISO balloting process to become an CD “committee draft”) into account. This ifcXML2 proposal has gained fuller acceptance and is the focus of the body of this document. Earlier versions of the schema are discussed in the appendix.

The RC2 of ifcXML2 has been published on Feb 05, 2005 in preparation of the ITM summit in Madrid, Feb 23.-24.2005. It has several bug fixes in the ifcXML2 language binding and translation and adds support for the IFC2x2 Addendum 1 schema. It has been improved to be parsed by the new/improved XML tools (like xmlspy2005).

The ifcXML2x2 schema was regenerated alongside the first release of the ifcXML2x3 schema in May 2007, using exactly the same options. This has ensured that both are fully interoperable. Refer to the change log for IFC2x3 and IFC2x3 TC1 for details of the internal changes to the IFC schema. These changes are equally reflected in ifcXML2x3.

IFC Release	ifcXML Release	Methodology	Notes
IFC2x	ifcXML2x	IAI's own EXPRESS to XSD conversion rules	Outdated, shall not be used any longer
IFC2x2			No official ifcXML release
IFC2x2 Add 1	ifcXML2x2	Part28 ed.2 CD level	RC2 now obsolete, final release published
IFC2x3	ifcXML2x3	Part28 ed.2 CD level	
IFC2x3 TC1	ifcXML2x3		Identical to IFC2x3 on EXPRESS level, therefore ifcXML2x3 is used for IFC2x3 TC1

Table 1: IFC and ifcXML releases

2.2 Definition

The content of an ifcXML file is constrained by the XML standard, the ISO-130303 part 28 edition 2 standard, the current ifcXML schema release and by requirements from the IFC schema and implementation guide. The effect of this is to create a recognizable and consistent representation that can be anticipated when reading and must be achieved when writing ifcXML models.

The following definition of terms in regard to ifcXML shall apply:

- **ifcXML methodology:** the methodology to automatically convert the IFC source definition (written in EXPRESS) into the ifcXML schema, and to convert data sets (provided as SPF or

EXPRESS database populations) into ifcXML documents (or files). The current methodology was developed in year 2002-4 to adopt the ongoing Part28 development for the generation of an ifcXML schema from the IFC EXPRESS source definition, the IFC2x edition2 version has been the first IFC version to which this methodology had been applied. The methodology is available at the IAI international website and currently relates to the CD version (spring 2004) of Part28.

- The second edition of Part28 is an ISO deliverable under ISO copyright. Therefore it cannot be published on a publicly accessible website. The IAI regrets this policy but has to conform to it.
- The second edition of Part28 (at time of writing) has been accepted as a Committee Draft by a successful balloting process. However the change requests that had been forwarded with the balloting will require amendments to the current document.
- As of May 2007 a new version of Part28 has been accepted by ISO. It is the Draft International Standard, DIS. Due to several remaining issues on the DIS version and the still very limited support by EXPRESS databases; it was decided to still work on the DIS version. Future releases of ifcXML may use the DIS or IS version of ISO10303-28 ed.2.
- The ifcXML specific part of the methodology is applied to Part28 as a configuration file (see).
- **ifcXML schema:** the XML schema definition that correlates to the IFC EXPRESS definition of a given IFC release and that had been generated by applying the ifcXML methodology.
 - The ifcXML schema is specific to the ifcXML methodology used (compare with Appendix 1 for the old ifcXML1 methodology) and the IFC EXPRESS version.
 - The most recent ifcXML schema is ifcXML for IFC2x3 (using the. Part28 2nd Ed. methodology).
- **ifcXML file:** an XML document that can be validated by an ifcXML schema.
 - Each ifcXML file shall be well structured and validate against a valid ifcXML schema.
 - The most recent ifcXML files are those that validate against the ifcXML for IFC2x3.

2.2.1 XML and XLINK

Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML ([ISO 8879](#)). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. (W3C)

The XLINK specification defines the XML Linking Language which allows elements to be inserted into XML documents in order to create and describe links between resources. It uses XML syntax to create structures that can describe links similar to the simple unidirectional hyperlinks of today's HTML, as well as more sophisticated links. (W3C)

2.2.2 XSD

XML Schema (XSD) is the schema definition language used in ifcXML to describe the structure of ifcXML files. The XSD is automatically generated from the IFC source definition, the IFC EXPRESS longform. The automatic generation is compliant to Part28 edition 2.

2.2.2.1 Part 28 edition 2

The generation of ifcXML from its source, the IFC EXPRESS specification is compliant to the general method to convert EXPRESS schemas into XML schemas. This method is developed under ISO TC184/SC4 as Part 28 edition 2:

Product data representation and exchange — Part 28: Implementation methods: XML representation of EXPRESS schemas and data

A detailed introduction into the translation method (the ifcXML methodology following Part28) is outside the scope of this document. The document can be obtained via the official ISO channels.

A short summary on Part28 edition 2

The scope of Part28 edition 2 is to provide a mapping between any EXPRESS schema and an XML schema (XSD) that semantics of which shall conform to the original EXPRESS schema. The syntax (or structure) of the XSD can be adjusted (within limitations) by providing a configuration file to the mapping. The resulting XSD shall provide some type safety (although it might be not as strict as EXPRESS).

In addition Part28 edition 2 provides some guidance to the generation of XML files, the container elements used, the name spaces, etc.

The following table shows the translation of a small snippet of IFC as an example:

Original IFC EXPRESS definition	ifcXML schema (generated by Part28)
<pre> ENTITY IfcProperty ABSTRACT SUPERTYPE OF (ONEOF (IfcComplexProperty ,IfcSimpleProperty)); Name : IfcIdentifier; Description : OPTIONAL IfcText; END_ENTITY; ENTITY IfcComplexProperty SUBTYPE OF (IfcProperty); UsageName : IfcIdentifier; HasProperties : SET [1:?] OF IfcProperty; END_ENTITY; <without where rules and inverse> <other subtypes skipped> </pre>	<pre> <xs:element name="IfcProperty" type="ifc:IfcProperty" abstract="true" substitutionGroup="ex:Entity" nillable="true"/> <xs:complexType name="IfcProperty" abstract="true"> <xs:complexContent> <xs:extension base="ex:Entity"> <xs:sequence> <xs:element name="Name" type="ifc:IfcIdentifier"/> <xs:element name="Description" type="ifc:IfcText" nillable="true" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> <xs:element name="IfcComplexProperty" type="ifc:IfcComplexProperty" substitutionGroup="ifc:IfcProperty" nillable="true"/> <xs:complexType name="IfcComplexProperty"> <xs:complexContent> <xs:extension base="ifc:IfcProperty"> <xs:sequence> <xs:element name="UsageName" type="ifc:IfcIdentifier"/> <xs:element name="HasProperties"> <xs:complexType> <xs:sequence> <xs:element ref="ifc:IfcProperty" maxOccurs="unbounded"/> </xs:sequence> <xs:attribute ref="ex:itemType" fixed="ifc:IfcProperty"/> <xs:attribute ref="ex:cType" fixed="set"/> <xs:attribute ref="ex:arraySize" use="optional"/> </xs:complexType> </xs:element> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

In addition an `ex:Entity` base type is introduced in Part28 for all element and `complexType`.

```
<xs:element name="Entity" type="ex:Entity"/>
<xs:complexType name="Entity" abstract="true">
  <xs:attribute name="href" type="xs:anyURI"
    use="optional"/>
  <xs:attribute name="ref" type="xs:IDREF"
    use="optional"/>
  <xs:attribute name="proxy" type="xs:IDREF"
    use="optional"/>
  <xs:attribute name="edo" type="xs:anyURI"
    use="optional"/>
  <xs:attributeGroup ref="ex:instanceAttributes"/>
</xs:complexType>

<xs:attributeGroup name="instanceAttributes">
  <xs:attribute name="id" type="xs:ID"
    use="optional"/>
  <xs:attribute name="path" type="xs:NMTOKENS"
    use="optional"/>
  <xs:attribute name="pos" use="optional">
    <xs:simpleType>
      <xs:restriction>
        <xs:simpleType>
          <xs:list itemType="xs:integer"/>
        </xs:simpleType>
        <xs:minLength value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
```

Part28 defines a container structure for all XML documents, the outermost container element is `iso_10303_28` that may include:

- one header, the `iso_10303_28_header` element (used in ifcXML),
- one or more unit(s) of serialization, the `uos` element (a single `uos` is used in ifcXML),
- in addition the actual configuration, the reference to the EXPRESS schema, and the original EXPRESS schema population but those are not accepted for ifcXML.

An ifcXML document shall therefore contain

- exactly one root element `iso_10303_28` within the name space
 - `urn:iso.org:standard:10303:part(28):version(2):xmlschema:common`
 - name space prefix (if used) `ex`
- within `iso_10303_28` zero or one child element `iso_10303_28_header` within the name space
 - `urn:iso.org:standard:10303:part(28):version(2):xmlschema:common`
 - name space prefix (if used) `ex`
- within `iso_10303_28` exactly one child element `uos` within the name space
 - `http://www.iai-tech.org/ifcXML/IFC2x3/FINAL` (IFC schema specific)
 - name space prefix (if used) `ifc`

2.2.2.2 Part28 configuration file

The language binding between the EXPRESS language and the XSD is not always deterministic, there are areas where the same EXPRESS structure can be mapped into different XSD structures while keeping the same semantic content. Those options are captured by the configuration file that instructs a translator on how to convert the EXPRESS structure into the XSD structure. This configuration file can for example be loaded into the EPM Jotne Express Data Manager.

```
<?xml version="1.0" encoding="UTF-8"?>
<ex:configuration
  xmlns:ex="urn:iso.org:standard:10303:part(28):version(2):xmlschema:common"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    urn:iso.org:standard:10303:part(28):version(2):xmlschema:common
    http://www.iai-tech.org/ifcXML/Ifc2x3/FINAL/configuration.xsd"
  ex:id="i-ifc2x3">
  <ex:option
    ex:inheritance="true"
    ex:exp-type="unspecified"
    ex:exp-attribute="double-tag"
    ex:flatten="false"
    ex:sparse="true"
    ex:tagless="false"
    ex:naming-convention="preserve-case"/>
  <ex:type ex:select="NUMBER" ex:map="xs:double"/>
</ex:configuration>
```

2.2.2.3 Explanation of the configuration settings

The configuration settings determine the exact syntax of the ifcXML schema. There are chosen to provide a maximum of type safety and accessibility of individual elements (e.g. the preference to have a sequence of elements, instead of an xs:list data type for 3 coordinates).

- `naming-convention="preserve-case"`
The XML names correspond to the EXPRESS names without changing capitalization.
- `exp-type="unspecified"`
elements corresponding to EXPRESS named data types can appear as root elements and contained elements (no restriction of what can be referenced or contained)
- `exp-attribute="double-tag"`
An attribute with entity data type is represented by a pair of nested XML elements: one for the attribute, and one for the entity that populates it.
- `tagless="false"`
Aggregate values shall be represented by the list of instance elements form (each aggregate element mapped to an XML element) and not by xs:list for simple data types.
- `inheritance="true"`
The simple inheritance structure of the EXPRESS schema shall be preserved in XSD (by substitutionGroup and complexContent).
- `flatten="false"`
not relevant for ifcXML – refers to the flattening of multi-dimensional aggregates, that do not exist in the IFC EXPRESS schema.
- `sparse="true"`
not relevant for ifcXML – refers to the handling of optional arrays (about the missing “NIL” values – whether exchanged by empty element (xsi:nil) or by forcing all to have position attributes. The IFC EXPRESS schema does not have optional arrays.
- `keep="true"`
This refers to derived and inverse attributes. This attribute has not been included but may be useful for working with partial model exchange. Inverse attributes would be supported like the other attributes but marked as optional and nillable.

2.3 XML Documents

In EXPRESS STEP based IFC implementations the actual exchange file has the syntax following ISO10303-21 (the STEP physical file, SPF) format. The current IFC documents, such as the IFC implementation guide, contain examples given in the SPF format.

The ifcXML document is the exchange file format for ifcXML and should be validated against the ifcXML schema. Each data element that can be exchanged within a SPF can also be exchanged within an ifcXML document.

There is some flexibility within the ifcXML schema that allows a rather unconstrained composition of XML documents. Further guidelines may provide additional restrictions.

Optional attributes (or attributes of 0..1 cardinality), if omitted, are represented by a '\$' in SPF, in ifcXML it can either be omitted (due to minOccurs="0" particle) or given as empty element (due to nillable="true" particle)

IFC SPF format data sample	IFC ifcXML format data sample
<pre>#84 = IFCPROPERTYINGLEVALUE ('Red', \$, IFCINTEGER(255), \$); using omitting of not specified optional attributes (using empty elements (</pre>	<pre><IfcPropertySingleValue id="i84"> <Name>Red</Name> <NominalValue> <IfcInteger>255</IfcInteger> </NominalValue> </IfcPropertySingleValue> <IfcPropertySingleValue id="i84"> <Name>Red</Name> <Description/> <NominalValue> <IfcInteger>255</IfcInteger> </NominalValue> <Unit/> </IfcPropertySingleValue></pre>

References to other entity instances (attributes or relationships with an entity data type) are always represented by reference in SPF (#123 notation), in ifcXML it can either be contained or referenced (due to nillable="true" particle), in the latter case it is required that the empty element has a 'ref' or 'href' attribute.

IFC SPF format data sample	IFC ifcXML format data sample
<pre>#84=IFCPROPERTYINGLEVALUE ('Red', \$, IFCINTEGER(255), \$); #85=IFCPROPERTYINGLEVALUE ('Green', \$, IFCINTEGER(0), \$); #86=IFCPROPERTYINGLEVALUE ('Blue', \$, IFCINTEGER(0), \$); #87=IFCCOMPLEXPROPERTY ('Color', \$, 'Color', (#84,#85,#86)); using containment →</pre>	<pre><IfcComplexProperty id="i87"> <Name>Color</Name> <UsageName>Color</UsageName> <HasProperties> <IfcPropertySingleValue> <Name>Red</Name> <NominalValue> <IfcInteger>255</IfcInteger> </NominalValue> </IfcPropertySingleValue> <IfcPropertySingleValue> <Name>Green</Name> <NominalValue> <IfcInteger>0</IfcInteger> </NominalValue> </IfcPropertySingleValue> <IfcPropertySingleValue> <Name>Blue</Name> <NominalValue> <IfcInteger>0</IfcInteger> </NominalValue> </IfcPropertySingleValue> </HasProperties> </IfcComplexProperty></pre>

<p style="color: blue; margin: 0;">using reference →</p>	<pre> <IfcComplexProperty id="i87"> <Name>Color</Name> <UsageName>Color</UsageName> <HasProperties> <IfcPropertySingleValue xsi:nil="true" href="i84"/> <IfcPropertySingleValue xsi:nil="true" href="i85"/> <IfcPropertySingleValue xsi:nil="true" href="i86"/> </HasProperties> </IfcComplexProperty> <IfcPropertySingleValue id="i84"> <Name>Red</Name> <NominalValue> <IfcInteger>255</IfcInteger> </NominalValue> </IfcPropertySingleValue> <IfcPropertySingleValue id="i85"> <Name>Green</Name> <NominalValue> <IfcInteger>0</IfcInteger> </NominalValue> </IfcPropertySingleValue> <IfcPropertySingleValue id="i86"> <Name>Blue</Name> <NominalValue> <IfcInteger>0</IfcInteger> </NominalValue> </IfcPropertySingleValue> </pre>
----------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.3.1 Entities

All entities within the content section have `ifc` as a prefix identical to their EXPRESS identifiers.

NOTE: in the obsolete ifcXML1 the ifc prefix had been stripped from the entity name, this is no longer the case.

2.3.2 Namespace

Entities may use a namespace, normally using the name space prefix “ifc:”. In order to ensure interoperable XML documents that use multiple namespaces, the namespaces themselves are required to be named uniquely. It is a common usage convention to use a common globally unique identifier string that is composed of a URL that is “owned” by the organization developing the schema. For the IAI, all namespaces belong to a common “root” URI, specifically “http://www.iai-tech.org/ifcXML”. To the end of this common root, a short identifier is put at the end separated by a forward slash. For example, the “ifc” namespace would have a full unique identifier of “http://www.iai-tech.org/ifcXML/IFC2x3/FINAL”. This string then uniquely identifies the namespace, which ensures there is no conflict with other industries and schema data definitions. By convention, we assign the shorthand prefix tag “ifc:” is typically assigned in a schema declaration to mean the same thing as the URL so that the XML files are much more human readable, yet maintain uniqueness for the XML parsing program. The file suffix, namespace prefix and full namespace qualifier will often, for convenience and simplicity, be treated as synonymous.

There are three examples of how to use name spacing in ifcXML documents. Of the three examples, the second has the benefit of confining the namespace declarations to the first element in the file, with all subsequent elements being straight-forward. For this reason case 2 is preferable.

Note that the equivalent files for ifc2x2 can be created by direct substitution of the string 'ifc2x3', and then noting that the schema is called ifc2x2_final.xsd. (compared against ifc2x3.xsd).

2.3.2.1 Using default namespaces for both the iso_10303_28 and uos elements

```

<iso_10303_28
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns=
"urn:iso.org:standard:10303:part(28):version(2):xmlschema:common"
  xsi:schemaLocation=
    "urn:iso.org:standard:10303:part(28):version(2):xmlschema:common
    http://www.iai-tech.org/ifcXML/IFC2x3/FINAL/ex.xsd"
  version="2.0"
>
  <iso_10303_28_header>
    <name>Case 1, using default namespaces for both the iso_10303_28
      and uos elements</name>
    <time_stamp>2007-05-01T17:07:21</time_stamp>
    <author>Thomas Liebich and Nick Nisbet</author>
    <organization>IAI Model Support Group</organization>
    <preprocessor_version>2007 sp3</preprocessor_version>
    <originating_system>Altova XMLSPY</originating_system>
    <authorization>IAI Model Support Group</authorization>
    <documentation>refer to IAI ifcXML implementation
      guide</documentation>
  </iso_10303_28_header>
  <uos id="uos_1" description="" configuration="i-ifc2x3" edo=""
    xmlns="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL"
    xmlns:ex=
"urn:iso.org:standard:10303:part(28):version(2):xmlschema:common"
    xsi:schemaLocation="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL
    http://www.iai-tech.org/ifcXML/IFC2x3/FINAL/IFC2X3.xsd"
  >
    <IfcOrganization id="i1504">
      <Name>IAI Model Support Group</Name>
    </IfcOrganization>
    <IfcCartesianPoint id="i505">
      <Coordinates>
        <IfcLengthMeasure>3600.0</IfcLengthMeasure>
        <IfcLengthMeasure>0.0</IfcLengthMeasure>
        <IfcLengthMeasure>0.0</IfcLengthMeasure>
      </Coordinates>
    </IfcCartesianPoint>
    <IfcDirection id="i506">
      <DirectionRatios>
        <ex:double-wrapper>0.</ex:double-wrapper>
        <ex:double-wrapper>0.</ex:double-wrapper>
        <ex:double-wrapper>1.</ex:double-wrapper>
      </DirectionRatios>
    </IfcDirection>
  </uos>
</iso_10303_28>

```

2.3.2.2 Using default namespace for the uos element (Preferred)

```

<?xml version="1.0" encoding="UTF-8"?>
<ex:iso_10303_28
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ex=
    "urn:iso.org:standard:10303:part(28):version(2):xmlschema:common"
  xmlns="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL"
  xsi:schemaLocation="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL
    http://www.iai-tech.org/ifcXML/IFC2x3/FINAL/IFC2x3.xsd"
  version="2.0"
>
  <ex:iso_10303_28_header>
    <ex:name>Case 2, using default namespace for the uos
      element</ex:name>
    <ex:time_stamp>2007-05-01T17:07:21</ex:time_stamp>
    <ex:author>Thomas Liebich and Nick Nisbet</ex:author>
    <ex:organization>IAI Model Support Group</ex:organization>
    <ex:preprocessor_version>2007 sp3</ex:preprocessor_version>
    <ex:originating_system>Altova XMLSPY</ex:originating_system>
    <ex:authorization>IAI Model Support Group</ex:authorization>
    <ex:documentation>refer to IAI ifcXML implementation
      guide</ex:documentation>
  </ex:iso_10303_28_header>
  <uos id="uos_1" description="" configuration="i-ifc2x3" edo="">
    <IfcOrganization id="i1504">
      <Name>IAI Model Support Group</Name>
    </IfcOrganization>
    <IfcCartesianPoint id="i505">
      <Coordinates>
        <IfcLengthMeasure>3600.0</IfcLengthMeasure>
        <IfcLengthMeasure>0.0</IfcLengthMeasure>
        <IfcLengthMeasure>0.0</IfcLengthMeasure>
      </Coordinates>
    </IfcCartesianPoint>
    <IfcDirection id="i506">
      <DirectionRatios>
        <ex:double-wrapper>0.</ex:double-wrapper>
        <ex:double-wrapper>0.</ex:double-wrapper>
        <ex:double-wrapper>1.</ex:double-wrapper>
      </DirectionRatios>
    </IfcDirection>
  </uos>
</ex:iso_10303_28>

```

2.3.2.3 Not using default namespaces

```

<?xml version="1.0" encoding="UTF-8"?>
<ex:iso_10303_28
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ifc="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL"
  xmlns:ex=
    "urn:iso.org:standard:10303:part(28):version(2):xmlschema:common"
  xsi:schemaLocation=
    "urn:iso.org:standard:10303:part(28):version(2):xmlschema:common
    http://www.iai-tech.org/ifcXML/IFC2x3/FINAL/ex.xsd"
  version="2.0"
>
  <ex:iso_10303_28_header>
    <ex:name>Case 3, not using default namespaces</ex:name>
    <ex:time_stamp>2007-05-01T17:07:21</ex:time_stamp>
    <ex:author>Thomas Liebich and Nick Nisbet</ex:author>
    <ex:organization>IAI Model Support Group</ex:organization>
    <ex:preprocessor_version>2007 sp3</ex:preprocessor_version>
    <ex:originating_system>Altova XMLSPY</ex:originating_system>
    <ex:authorization>IAI Model Support Group</ex:authorization>
    <ex:documentation>refer to IAI ifcXML implementation
      guide</ex:documentation>
  </ex:iso_10303_28_header>
  <ifc:uos id="uos_1" description="" configuration="i-ifc2x3" edo=""
    xmlns="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL"
    xsi:schemaLocation="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL
    http://www.iai-tech.org/ifcXML/IFC2x3/FINAL/IFC2x3.xsd"
  >
    <IfcOrganization id="i1504">
      <Name>IAI Model Support Group</Name>
    </IfcOrganization>
    <IfcCartesianPoint id="i505">
      <Coordinates>
        <IfcLengthMeasure>3600.0</IfcLengthMeasure>
        <IfcLengthMeasure>0.0</IfcLengthMeasure>
        <IfcLengthMeasure>0.0</IfcLengthMeasure>
      </Coordinates>
    </IfcCartesianPoint>
    <IfcDirection id="i506">
      <DirectionRatios>
        <ex:double-wrapper>0.</ex:double-wrapper>
        <ex:double-wrapper>0.</ex:double-wrapper>
        <ex:double-wrapper>1.</ex:double-wrapper>
      </DirectionRatios>
    </IfcDirection>
  </ifc:uos>
</ex:iso_10303_28>

```

2.3.3 Properties

- **id, href/ref**

Entities may have an 'id' attribute or either a 'ref' or 'href' attribute or neither. 'id', 'ref' and 'href' attributes must be non-numeric. An entity with an 'ref' or 'href' should have no children but should have the `xsi:nil="true"` attribute: this stops validation processes objecting to the absence of mandatory elements below.

Any entity with a local 'ref' or 'href' should correspond to another entity of the same type with the matching 'id' attribute. The 'ref' attribute shall be used, when the referenced element with the matching 'id' attribute resides in the same `uos` element. The 'href' attribute shall be used when the referenced element exists in an external source. It is illegal to use both, 'ref' and 'href' in the same element. Any entity with a remote 'href' need not correspond within the ifcXML document.

```
<IfcBuildingStorey id="bs1">
  <GlobalId>3jXakT2cDAD9uccVPFcs$8</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true"
      href="http://www.server.com/project1/#i2000"/>
  </OwnerHistory>
  <Name>First floor</Name>
  <ObjectPlacement>
    <IfcLocalPlacement xsi:nil="true" ref="i2001"/>
  </ObjectPlacement>
  <CompositionType>element</CompositionType>
  <Elevation>3150.0</Elevation>
</IfcBuildingStorey>
```

- **selects**

Where the schema allows a choice of entity type, this can be done directly. The use of the attribute to indicate the choice is optional.

- **real**

Real numbers may be in exponential format

```
<IfcLengthMeasure>100.0E-3</IfcLengthMeasure>
```

- **text**

Text is unquoted, but the sensitive characters in XML should be escaped for example

< is represented as <

- **lists, sets and bags**

Lists are given with distinct elements for each item. The use of the attribute to identify the type of list and/or the number of elements is optional. In addition each element within the sequence representing the list, set or bag may have a position attribute.

```
<IfcCartesianPoint id="i1001">
  <Coordinates ex:cType="list">
    <IfcLengthMeasure pos="0">100.0</IfcLengthMeasure>
    <IfcLengthMeasure pos="1">100.0</IfcLengthMeasure>
    <IfcLengthMeasure pos="2">100.0</IfcLengthMeasure>
  </Coordinates>
</IfcCartesianPoint>
```

- **Boolean**

"true" and "false" can appear as values for Boolean values

▪ Date representations

There are two date representations used within the ifcXML schema.

- (1) Seconds since 1 Jan 1970 for example

```
<CreationDate>1086769501</CreationDate>
```

- (2) Structured date with year, month and day indicated separately for example

```
<EditionDate>
  <IfcCalendarDate>
    <DayComponent>01</DayComponent>
    <MonthComponent>01</MonthComponent>
    <YearComponent>1991</YearComponent>
  </IfcCalendarDate>
</EditionDate>
```

2.4 Limitations

2.4.1 Known limitations

An ifcXML document will not contain three types of information:

- Inverse relationships: a relationship in an EXPRESS schema can imply its inverse, so that applications can back-track as well as advance through the data.
- Derived attributes, Again an EXPRESS schema can include procedural methods to derive attributes such as the transformation matrixes for placements.
- An ifcXML document is not constrained by the schema to comply with constraints implemented in the EXPRESS schema, including the 'WHERE' rules that are added to entities to restrict the range or combinations of attribute values. Applications generating ifcXML should ensure that these constraints are not violated.

2.4.2 Possible application limitations

If the ifcXML schema is used to generate a model in a database implementation that is not compliant to the EXPRESS standard then certain conflicts or limitations may be encountered. A few potential pitfalls are indicated here but the list cannot be definitive. If necessary the interface to the database may need to protect the database from the conflict. In some cases a simple name mapping may be sufficient. These conflicts can occur in case sensitive and case insensitive applications.

2.4.2.1 Id (Identity) and id (Xlink)

The attribute 'Id' or 'ID' could conflict with the 'id' used in XLINK. The following entities are affected:

```

IfcPerson
IfcOrganisation
IfcProjectOrder
IfcCostSchedule
```

This workaround can be introduced manually, into the XSD, or SQL DDL script.

The second entry below is offered as a plausible and systematic renaming, pre-pending 'the'. An alternative might be an underscore '_'.

```

'Id' ('theId')
'ID' ('anotherID')
```

NOTE: One relational-object hybrid database automatically added a third, hidden, 'id' field. In this case the XLINK attribute 'id' had to be renamed 'xmlId'.

2.4.2.2 SQL keywords

Some SQL keywords may conflict with attribute names in the IFC schema. A workaround can be introduced manually, into the XSD, or SQL DDL script. A suggested renaming would be to prefix 'the' or '_' before. Alternatively it may be necessary to ensure that all SQL queries and updates use quoted field names. If the SQL is automatically generated, this may not be possible.

- 'Role'
- 'User'
- 'Outer'
- 'Values'
- 'Rows'
- 'Position'

2.4.2.3 Long entity names

This is not a problem with all RDMS applications, but four entity names in particular are over 32 characters. [IfcCartesianTransformationOperator*](#) could be mapped to [IfcTransformation*](#).

- [IfcCartesianTransformationOperator2D](#)
- [IfcCartesianTransformationOperator3D](#)
- [IfcCartesianTransformationOperator2DnonUniform](#)
- [IfcCartesianTransformationOperator3DnonUniform](#)

2.4.2.4 Repeated attribute names

Caution may be needed if the database represents attributes as separate tables. In the IFC schema it is common for the same attribute name to be attached to distinct entities, often with a different data type.

For example 'Status' occurs frequently as an optional [IfcLabel](#) but also occurs once as an optional [IfcDocumentStatusEnum](#).

2.5 Current implementations of ifcXML

As of July 2004, there are a small number of applications known to be capable of using ifcXML. In general the distinction is between so-called "toolbox" or "server" implementations and "end-user" or "client" software.

This section lists the general toolbox implementations that can be used for generating software or other utilities that can handle ifcXML. The first two are model servers capable of working with both SPF (Part 21) and ifcXML (Part 28) files. The next two are general XML workbench applications capable of generating sample documents. In addition there are several commercial and prototype implementations of software using ifcXML (either directly, or through the use of toolboxes). These implementations are briefly discussed below.

It is expected that all these applications listed will converge to an exact implementation of the standard in the coming months.

2.5.1 EPM Express Data Manager v4.7 to v5.0

An EXPRESS and SDAI based database that supports Part28 conversion of populated data models, it accepts the ifcXML configuration file and therefore allows to reads and writes ifcXML

2.5.2 Eurostep Model Server (EMS)

An EXPRESS and SDAI based server that supports Part28 conversion of populated data models, it accepts the ifcXML configuration file but currently only allows the writing of ifcXML and not the reading.

2.5.3 XMLSPY 2007

It is a general XML tool that supports editing, parsing, converting, etc. of XML schema and documents. It has no awareness of Part28 and ifcXML per se, but maybe used to validate and further process ifcXML documents. The latest version of XMLSPY has a greatly improved and tighter validation against the schema.

Specifics:

2.5.4 TurboXML

It is a general XML tool that supports editing, parsing, converting, etc. of XML schema and documents. It has no awareness of Part28 and ifcXML per se, but maybe used to validate and further process ifcXML documents.

2.5.5 Octaga Modeller

It is a general viewer that allows to view geographic data (gml-based) and building data (ifc-based), for the viewing of building data it accepts both IFC-SPF and ifcXML.

- An XML streaming viewer
- Also supports GML for geographic data

2.5.6 ArchiCAD v9.0 to v11.0

ArchiCAD 9.0 has an EDMIFC_2x2 and 2x3 add-on, that can be downloaded. The add-ins have commands for saving and loading ifcXML files, and it can update the existing ArchiCAD model's IFC properties from an external ifcXML file. The add-on is based on EPM Express Data Manager and therefore has the same XML specifics as described in 2.5.1.

2.5.7 AEC3 Ltd

Two tools that incorporate ifcXML capabilities are currently under development:

- PIP Product Information Packager for product and property catalogues
- Dad AutoCAD ADT Drag-And-Drop tools for Schedule Data development

2.5.8 Research Center of Karlsruhe, Germany

The Research Center of Karlsruhe develops several tools supporting IFC based integration, including a viewer, the IfcStoreyView. The latest versions since version 1.5 have an ifcXML read capability. The functionality is provided by the underlying ECCO tool from PDTec, Germany, for more see 2.5.12.

2.5.9 TNO, Netherlands

The building IT department of TNO develops several tools supporting IFC based integration, including a viewer, the IfcEngine. An ifcXML support is currently under development.

2.5.10 ifcSVR300

The SECOM-developed ifcSVR300 has the ability to write IFCXML files, but the file has a number of minor errors.

2.5.11 ALLPLAN2006.2

ALLPLAN IFC converter since version ALLPLAN2006.2 can export and import ifcXML data files. The add-on is based on EPM Express Data Manager and therefore has the same XML specifics as described in 2.5.1.

2.5.12 ECCO (EXPRESS toolbox by PDTec)

The ECCO Express toolbox has the principle functionality to support an ISO10303-28 ed.2 representation of data according to an Express schema, however small differences to EPM's implementation of ISO10303-28 ed.2 have been detected.

3 Use cases

This section suggests a methodology for implementing ifcXML. The use cases are not fully developed and merely act as introductions to the examples, which in turn are for illustration only.

The basic steps for setting up an XML import/export interface in your application software:

1. Determine what data needs to be read into or written from your software application to support one or more usage scenarios that are of interest to your users and understand what data you may want to transfer on behalf of your users. Talk to your users about the data they would like to transfer to/from your application.
2. Review the IFC schema to see which entity definitions are relevant.
3. Understand how the information is stored in your own application and what programming interfaces are required to get and put information into application storage.
4. Design and document a mapping table between the data structures in your own application for the data you want to transfer (either import, export or both). In one column of the mapping table define the storage location and function calls required to get or put information from/to the application storage. In a second column define the Xpath to the element in the ifcXML schemas that match the data item(s) you want to transfer. Xpath is a notation that can be used by XML Stylesheet Language Transformations (XSLT) to “flatten” the XML tree structure in an XML instance file by providing a unique path definition to data contained within an XML instance file.
5. Write application code for “import” and/or “export” that uses an XML parser, such as MSXML or Xerces, and the DOM or SAX application programming interface(s) to parse the XML instance file to get or put information into an ifcXML instance file and your application storage according to the mapping table you defined in step 4.

(based on “Using XML schemas for facilities equipment” FIATECH 2004)

6. Ensure that the applications appreciate and work according to the IFC Implementation Guide, particularly for:
 - ‘GlobalId’s and naming
 - Units and conversion factors
 - Geometry and axes.
 - [IfcOwnerHistory](#) and change logging

3.1 Use case: Material catalogue

3.1.1 Description

The use case indicates the data content of a material catalogue transmission.

3.1.2 Purpose:

Material suppliers and accreditation bodies have extensive libraries of material and construction properties which are of value in product selection and in performance prediction. Using ifcXML enables such data to be transmitted and received with confidence.

3.1.3 Actors:

- Material suppliers
- Accreditation bodies
- Engineering analysts
- Design professionals

3.1.4 Transactions:

A Request for Information may be issued by a design professional and a competent body may reply with a detailed response.

3.1.5 Applications:

- CRM suites
- EIA suites
- Analysis applications
- Object based CAD applications

3.1.6 Data content

```
<IfcMaterial id="i9000">
  <Name>Concrete</Name>
</IfcMaterial>

<IfcThermalMaterialProperties>
  <Material>
    <IfcMaterial xsi:nil="true" href="i9000"/>
  </Material>
  <SpecificHeatCapacity>300</SpecificHeatCapacity>
  <BoilingPoint/>
  <FreezingPoint/>
  <ThermalConductivity/>
</IfcThermalMaterialProperties>

<IfcExtendedMaterialProperties>
  <Material>
    <IfcMaterial xsi:nil="true" href="i9000"/>
  </Material>
  <ExtendedProperties>
    <IfcPropertyEnumeratedValue>
      <Name>ClimateChangeImpact</Name>
      <Description>Climate Change Impact Evaluation</Description>
      <EnumerationValues>
        <IfcLabel>Medium</IfcLabel>
      </EnumerationValues>
    </IfcPropertyEnumeratedValue>
  </ExtendedProperties>
</IfcExtendedMaterialProperties>
```

```
<EnumerationReference>
  <IfcPropertyEnumeration>
    <Name>PEnum_ClimateChange</Name>
    <EnumerationValues>
      <IfcLabel>High</IfcLabel>
      <IfcLabel>Low</IfcLabel>
      <IfcLabel>Medium</IfcLabel>
      <IfcLabel>Other</IfcLabel>
      <IfcLabel>NotKnown</IfcLabel>
      <IfcLabel>Unset</IfcLabel>
    </EnumerationValues>
  </IfcPropertyEnumeration>
</EnumerationReference>
</IfcPropertyEnumeratedValue>
</ExtendedProperties>
<Description>Whole Life Cycle Properties for Concrete
</Description>
<Name>Pset_WLC: Concrete</Name>
</IfcExtendedMaterialProperties>
```

3.1.7 Comment

The rating of materials could equally be implemented using [IfcMaterialClassification](#).

3.2 Use case: Bill of Quantities report

3.2.1 Description

This use case indicates the data content of a bill of quantities transmission.

3.2.2 Purpose:

A project model may be analyzed to generate a Bill of Quantities for the purposes of cost estimation and tendering. The quantity information, bill structure and cost data can be developed separately.

3.2.3 Actors:

- Quantity experts
- Tendering contractors

3.2.4 Transactions:

A Request for Tender may be accompanied by a specific breakdown of the model of the project. The response will add distinct information, which can be held in the model along with other tenders.

3.2.5 Applications:

- Construction Enterprise Integration Systems (EIS) or ERM suites

3.2.6 Data content

<!-- Quantities: associating basic measurements to the building elements -->

```
<IfcRelDefinesByProperties>
  <GlobalId>2YAqEv_AP2vuyLii8Lf0V0</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" href="i9075"/>
  </OwnerHistory>
  <RelatedObjects>
    <IfcWallStandardCase xsi:nil="true" href="i9163"/>
    <IfcWallStandardCase xsi:nil="true" href="i9251"/>
  </RelatedObjects>
  <RelatingPropertyDefinition>
    <IfcElementQuantity>
      <GlobalId>3qsqH3lvbFjPShG$jCakxE</GlobalId>
      <OwnerHistory>
        <IfcOwnerHistory xsi:nil="true" href="i9075"/>
      </OwnerHistory>
      <Name>Wall Quantities</Name>
      <!-- geometric mesasures are not locale specific -->
      <MethodOfMeasurement>geometry</MethodOfMeasurement>
      <Quantities>
        <IfcQuantityVolume>
          <Name>Volume</Name>
          <VolumeValue>7.551478</VolumeValue>
        </IfcQuantityVolume>
      </Quantities>
    </IfcElementQuantity>
  </RelatingPropertyDefinition>
</IfcRelDefinesByProperties>
```

```
<!-- Bill of Quantities: grouping elements into hierarchical sets -->
```

```
<IfcCostSchedule id="CS_001">
  <GlobalId>CS34567890123456789012</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" href="i9075"/>
  </OwnerHistory>
  <Name>BOQ</Name>
  <Description>Bill of Quantities Example</Description>
  <ID>BOQ-20040817</ID>
  <PredefinedType>userdefined</PredefinedType>
</IfcCostSchedule>
```

```
<IfcRelSchedulesCostItems>
  <GlobalId>RCS4567890123456789012</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" href="i9075"/>
  </OwnerHistory>
  <RelatedObjects>
    <IfcCostItem xsi:nil="true" href="CI_001"/>
  </RelatedObjects>
  <RelatingControl>
    <IfcCostSchedule xsi:nil="true" href="CS_001"/>
  </RelatingControl>
</IfcRelSchedulesCostItems>
```

```
<IfcCostItem id="CI_001">
  <GlobalId>CI34567890123456789012</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" href="i9075"/>
  </OwnerHistory>
  <Name>CI_001</Name>
  <Description>Internal wall costs</Description>
</IfcCostItem>
```

```
<IfcRelSchedulesCostItems>
  <GlobalId>RCS4567890123456789013</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" href="i9075"/>
  </OwnerHistory>
  <RelatedObjects>
    <IfcWallStandardCase xsi:nil="true" href="i9163"/>
    <IfcWallStandardCase xsi:nil="true" href="i9251"/>
  </RelatedObjects>
  <RelatingControl>
    <IfcCostItem xsi:nil="true" href="CI_001"/>
  </RelatingControl>
</IfcRelSchedulesCostItems>
```

```
<!-- Tender cost values added to the building elements -->
```

```
<IfcRelAssociatesAppliedValue>
  <GlobalId>RAV4567890123456789012</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" href="i9076"/>
  </OwnerHistory>
  <RelatedObjects>
    <IfcWallStandardCase xsi:nil="true" href="i9163"/>
    <IfcWallStandardCase xsi:nil="true" href="i9251"/>
  </RelatedObjects>
  <RelatingAppliedValue>
```

```

    <IfcCostValue xsi:nil="true" href="CV_001"/>
  </RelatingAppliedValue>
</IfcRelAssociatesAppliedValue>

<IfcCostValue id="CV_001">
  <Name>CV1</Name>
  <Description>Cost Value One</Description>
  <AppliedValue>
    <IfcMonetaryMeasure>199.99</IfcMonetaryMeasure>
  </AppliedValue>
  <CostType>Estimated Cost</CostType>
</IfcCostValue>

<!-- Tender calculation elaborating the quoted cost value -->

<IfcAppliedValueRelationship>
  <ComponentOfTotal>
    <IfcCostValue xsi:nil="true" href="CV_001"/>
  </ComponentOfTotal>
  <Components>
    <IfcCostValue>
      <Name>CV1a</Name>
      <Description>Basic cost</Description>
      <AppliedValue>
        <IfcMonetaryMeasure>242.41</IfcMonetaryMeasure>
      </AppliedValue>
      <CostType>Estimated Cost</CostType>
    </IfcCostValue>
    <IfcCostValue>
      <Name>CV1b</Name>
      <Description>Trade discount</Description>
      <AppliedValue>
        <IfcRatioMeasure>0.825</IfcRatioMeasure>
      </AppliedValue>
      <CostType>Estimated Cost</CostType>
    </IfcCostValue>
  </Components>
  <ArithmeticOperator>multiply</ArithmeticOperator>
</IfcAppliedValueRelationship>

```

3.2.7 Comment

The tendering contractor does not have to use the same breakdown of the model as the quantifier or the reporter of the Bill of Quantities as each uses distinct groupings of individual objects in the model.

3.3 Use case: Adding user defined properties

3.3.1 Description.

This use case examines the inclusion of extra information using property sets

3.3.2 Purpose:

To communicate the detail of a building so that each building storey can be graded for fire assessment, and to receive the determination back

3.3.3 Actors:

- Designer
- Fire regulatory authority

3.3.4 Transactions:

The process of assessing the classification need not be completed instantaneously, but the new information should be associated with the fire authority.

3.3.5 Applications:

- CAD or rule based applications

3.3.6 Data content

```
<IfcBuildingStorey id="bs1">
  <GlobalId>3jXakT2cDAD9uccVPFcs$8</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" href="arch1"/>
  </OwnerHistory>
  <Name>First floor</Name>
  <ObjectPlacement>
    <IfcLocalPlacement xsi:nil="true" href="i2001"/>
  </ObjectPlacement>
  <CompositionType>element</CompositionType>
  <Elevation>3150.0</Elevation>
</IfcBuildingStorey>

<IfcRelDefinesByProperties>
  <GlobalId>1fvr_57Xf6Rgaz_vrADYHZ</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" ref="fire1"/>
  </OwnerHistory>
  <RelatedObjects ex:cType="set">
    <IfcBuildingStorey ex:pos="0" xsi:nil="true" ref="bs1"/>
    <IfcBuildingStorey ex:pos="0" xsi:nil="true" ref="bs2"/>
  </RelatedObjects>
  <RelatingPropertyDefinition>
    <IfcPropertySet>
      <GlobalId>1lpBuQpwH20OGi7rxnE1xi</GlobalId>
      <OwnerHistory>
        <IfcOwnerHistory xsi:nil="true" ref="fire1"/>
      </OwnerHistory>
      <Name>PSet_BuildingStoreyCommon</Name>
    </IfcPropertySet>
  </RelatingPropertyDefinition>
</IfcRelDefinesByProperties>
```

```
<HasProperties ex:cType="set">
  <IfcPropertySingleValue>
    <Name>AboveGround</Name>
    <Description>
      Indication whether this building storey is fully above ground
      (TRUE), or below ground (FALSE), or partially above and below
      ground (UNKNOWN) - as in sloped terrain.
    </Description>
    <NominalValue>
      <IfcLogical>true</IfcLogicale>
    </NominalValue>
  </IfcPropertySingleValue>
</HasProperties>
</IfcPropertySet>
</RelatingPropertyDefinition>
</IfcRelDefinesByProperties>
```

3.3.7 Comment

The ownership history of the relationship and the property set will be different to the ownership history of the building storey itself.

The fire officer or an application has evaluated the building model and the site topography and judged that the building storey (and others) is graded as 'AboveGround' being 'true'. This information is communicated accurately back to the architect.

Information about Property sets and properties is available in the IAI IFC documentation. Some property sets are mandated by the IAI MSG and form part of the published schema.

4 Writing ifcXML

4.1 Mandatory content

The content of an ifcXML file is constrained by the XML standard, the part 28 edition 2 standard, the current ifcXML schema release and by requirements from the IFC schema and implementation guide. The effect of this is to create a recognizable and consistent representation that can be anticipated when reading and must be achieved when writing ifcXML models.

It should be noted that all entities and attribute names are case sensitive. The case usage follows that used by the EXPRESS schema and can be summarized as:

- Entities are prefixed **Ifc** and are CamelCased for example **IfcBuildingStorey**
- Attributes are not suffixed and are CamelCased for example 'CreationDate'

4.1.1 Mandatory header and footer

The content of an ifcXML file occurs within the part 28 **uos** element which carries the external schema references and namespace identifiers. ifcXML requires as the container element the **iso_10303_28** element, containing a single **iso_10303_28_header**, and one or several **uos** elements. It is suggested that there should be only one **uos** per document, for compatibility with part 21 and for simplicity of applications and change control.

Please refer to the three cases in 2.3.2.1-3 above

4.1.2 Mandatory elements

As a minimum, an ifcXML file should contain an **IfcProject** entity and its required attributes and related entities. These include a list of at least one default unit typically for length and at least one geometric representation context. These also include an **IfcOwnerHistory** entity which carries information on the origination of the **IfcProject**. The **IfcOwnerHistory** may well be referenced by other entities in the model. These requirements may not be relevant for partial models as long as they retain references to their source where the single **IfcProject** etc resides.

4.1.3 Mandatory attributes

Id

The Xlink 'id' attribute is optional but will be required to act as the targets for 'href' or 'ref' references. 'id' need not be present on entities to which no reference is made. These exceptions include contained elements and typically IFC Relationship entities.

Any identifier chosen must not be numeric, but an initial prefix 'i' or '_' (underscore) may be used. These identifiers are ephemeral and have no significance outside of the immediate context of the current document. No reliance can be made that they will repeat in a second document. When working with partial models, the 'href' entity may be used to refer to entities still residing on the source server.

GlobalId

Entities derived from **IfcRoot** which includes all spatial and building elements and relationships have a mandatory 'GlobalId' attribute. This is defined in the core documentation with suggested algorithms. It is expected to be a unique and unrepeatable 128 bit integer represented as a string of exactly 22 characters long, made up from the 26 character Western alphabet in upper and lower case, the ten numeric digits and the '\$' (dollar) and '_' (underscore) characters.

It is the responsibility of every application to persist GlobalId's when receiving entities and to generate original GlobalId's for new entities.

As an alternative implementation an application could use other information such as the originating company name and product identifier, padding to the required 22 characters. This is a risky implementation and may cause conflicts in the global information environment.

OwnerHistory

Entities derived from [IfcRoot](#) which includes all spatial and building elements and relationships have a mandatory 'OwnerHistory' attribute. The information held within this attribute is extensive and allows the data author, the time, and the application etc to be associated to the entity.

It is the responsibility of every application to create an [IfcOwnerHistory](#) when it adds to or modifies the model. The [IfcOrganization](#) and [IfcApplication](#) entities need be created only once.

4.2 Example minimal (empty) file

This example suggests the minimum content for an ifcXML file between the header and footer given above.

Note: Because various entities are nested within the hierarchy, they can not legitimately be referred to by other elements. An actual implementation might for example move the [IfcOwnerHistory](#) to the top level so as to allow other entities to refer to it. Indeed an implementation might move ALL entities to the top level and use only referencing and no nesting. This is discussed under 'Priority issues' below.

Note: Please review the ifc2x3 documentation for the full definition of [IfcProject](#) and related entities. For example strictly speaking a simpler [IfcRepresentationContext](#) could be used if no geometry was being transferred.

```
<IfcProject>
  <GlobalId>0NZgqcQj5AlvVyooSuXX4r</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory>
      <OwningUser>
        <IfcPersonAndOrganization>
          <ThePerson>
            <IfcPerson>
              <FamilyName>Nisbet</FamilyName>
            </IfcPerson>
          </ThePerson>
          <TheOrganization>
            <IfcOrganization>
              <Id>IAI</Id>
              <Name>International Alliance for Interoperability</Name>
            </IfcOrganization>
          </TheOrganization>
        </IfcPersonAndOrganization>
      </OwningUser>
      <OwningApplication>
        <IfcApplication>
          <ApplicationDeveloper>
            <IfcOrganization>
              <Id>AEC3</Id>
              <Name>AEC3 Ltd</Name>
            </IfcOrganization>
          </ApplicationDeveloper>
          <Version>0.0</Version>
          <ApplicationFullName>Application 0.0</ApplicationFullName>
        </IfcApplication>
      </OwningApplication>
    </IfcOwnerHistory>
  </OwnerHistory>
</IfcProject>
```

```

        <ApplicationIdentifier>Application</ApplicationIdentifier>
    </IfcApplication>
</OwningApplication>
    <ChangeAction>nochange</ChangeAction>
    <CreationDate>1086769501</CreationDate>
</IfcOwnerHistory>
</OwnerHistory>
<Name>ifcXML Project</Name>
<RepresentationContexts ex:cType="set">
    <IfcGeometricRepresentationContext>
        <ContextIdentifier>Plan</ContextIdentifier>
        <ContextType>Design</ContextType>
        <CoordinateSpaceDimension>3</CoordinateSpaceDimension>
        <WorldCoordinateSystem>
            <IfcAxis2Placement3D>
                <Location>
                    <IfcCartesianPoint>
                        <Coordinates>
                            <IfcLengthMeasure>0.0</IfcLengthMeasure>
                            <IfcLengthMeasure>0.0</IfcLengthMeasure>
                            <IfcLengthMeasure>0.0</IfcLengthMeasure>
                        </Coordinates>
                    </IfcCartesianPoint>
                </Location>
                <Axis>
                    <IfcDirection>
                        <DirectionRatios>
                            <ex:double-wrapper>0.0</ex:double-wrapper>
                            <ex:double-wrapper>0.0</ex:double-wrapper>
                            <ex:double-wrapper>1.0</ex:double-wrapper>
                        </DirectionRatios>
                    </IfcDirection>
                </Axis>
                <RefDirection>
                    <IfcDirection>
                        <DirectionRatios>
                            <ex:double-wrapper>1.0</ex:double-wrapper>
                            <ex:double-wrapper>0.0</ex:double-wrapper>
                            <ex:double-wrapper>0.0</ex:double-wrapper>
                        </DirectionRatios>
                    </IfcDirection>
                </RefDirection>
            </IfcAxis2Placement3D>
        </WorldCoordinateSystem>
    </IfcGeometricRepresentationContext>
</RepresentationContexts>
<UnitsInContext>
    <IfcUnitAssignment>
        <Units ex:cType="set">
            <IfcSIUnit>
                <UnitType>lengthunit</UnitType>
                <Prefix>milli</Prefix>
                <Name>metre</Name>
            </IfcSIUnit>
        </Units>
    </IfcUnitAssignment>
</UnitsInContext>
</IfcProject>

```

4.3 Types

4.3.1 Type matching using href and ref

It is obligatory that the type of all entities using a 'ref' or 'href' should match the type of the target with the matching 'id'. 'ref' should be used for referencing within the same `uos`, 'href' for referencing outside the `uos`. For example

```
<IfcOwnerHistory id="i60674">
  .
  .
  .
</IfcOwnerHistory xsi:nil="true" ref="i60674"/>
```

4.3.2 Bag, Set or List

Some attributes involve the repetition of an entity. The schema requires that the nature of the repetition be identified and reference should be made to the IFC documentation.

- Set: contains only one occurrence of any entity without any order.
- Bag: can contain more than one occurrence of any entity without any order.
- List: can contain more than one occurrence of any entity but the order is significant.

4.4 Order

The order of the `properties` within an entity is mandated by the IFC schema and follows the exactly same order as the SPF representation. Optional and derived properties may be absent, unlike in the SPF format where the \$ placeholder is used.

There is no specific order of the `elements` at the top level. No assumptions should be made when reading. There may be some benefit to the reader if the single `IfcProject` is written out first, but in general there are three different approaches which have been used. These are discussed below:

4.4.1 Alphabetical

Some model servers generate the ifcXML representation with all the top level entities sorted alphabetically. This may aid checking but will add to processing time during selective reading.

4.4.2 Dependency

Some applications may naturally generate entities in a hierarchical method starting with the topic of interest, and include the relationships, entities, and properties and resources necessary to describe it thereafter.

4.4.3 Reversed dependency

Some applications may naturally generate entities starting with the pre-requisite resources and properties before writing the entities and finally the relationships involved in a recursive method, starting with the topic of interest.

4.5 Priorities

When using ifcXML application developers will have different priorities and based on those priorities it may be possible to suggest slightly differing implementations.

4.5.1 Size

The most efficient ifcXML file size will arise when any entity that is referenced only once is nested ('contained') below the referencing element. This has the advantage of bringing together distinct units of meaning. With a detailed model, most building entities will rise to the top as they are referenced several times. Relationship entities will also appear at the top level, as they are

never referenced. For the convenience of the receiving system, the `IfcProject` (although usually only referenced once) should not be nested. Typically such a file will be about twice the size of the equivalent SPF.

However, by convention, geometric fundamentals such as points are only shared where they are fundamentally the same. For example, an `IfcBuilding` may be located at 0,0,0 relative to an `IfcSite`, and the main `IfcBuildingStorey` may be located at 0,0,0 within the `IfcBuilding`. The `IfcCartesianPoint` should not be shared, as the `IfcBuilding` may be relocated on the `IfcSite` and the datum for the `IfcBuilding` may not remain as that particular `IfcBuildingStorey`. Contrast this with a BREP representation of a cube made up of six faces. Each of the eight `IfcCartesianPoints` should be shared between three `IfcFaces` for efficiency and consistency.

```
<!--start of faces -->
<IfcFace>
  <Bounds>
    <IfcFaceOuterBound>
      <Bound>
        <IfcPolyLoop>
          <Polygon>
            <IfcCartesianPoint ref="q1" xsi:nil="true"/>
            <IfcCartesianPoint ref="q4" xsi:nil="true"/>
            <IfcCartesianPoint ref="q8" xsi:nil="true"/>
            <IfcCartesianPoint ref="q5" xsi:nil="true"/>
          </Polygon>
        </IfcPolyLoop>
      </Bound>
      <Orientation>true</Orientation>
    </IfcFaceOuterBound>
  </Bounds>
</IfcFace>
<!--and five other faces -->

<!--start of points -->
<IfcCartesianPoint id="q1">
  <Coordinates>
    <IfcLengthMeasure>0.500</IfcLengthMeasure>
    <IfcLengthMeasure>0.500</IfcLengthMeasure>
    <IfcLengthMeasure>0.500</IfcLengthMeasure>
  </Coordinates>
</IfcCartesianPoint>
<!--and seven other points -->
```

4.5.2 Simplicity

The simplest application may systematically write each entity in turn at the top level of the file and documenting all references using the 'id' to 'href' or 'ref' relationship. Some overhead may be taken to ensure that all important references are satisfied if only a partial model is being reported.

```
<IfcPerson id="i1004">
  <FamilyName>Steinmann</FamilyName>
  <GivenName>Rasso</GivenName>
</IfcPerson>
<IfcOrganization id="i1006">
  <Name>Nemetschek AG</Name>
</IfcOrganization>
<IfcPersonAndOrganization id="i1010">
  <ThePerson>
    <IfcPerson xsi:nil="true" ref="i1004"/>
```

```

</ThePerson>
<TheOrganization>
  <IfcOrganization xsi:nil="true" ref="i1006"/>
</TheOrganization>
</IfcPersonAndOrganization>

```

4.5.3 Speed

Where speed of execution at the receiving system is the primary concern, it may be acceptable to repeat the detail of any referenced element, allowing the receiving system to accept the input as a stream. This is dependant on the availability of large bandwidth, as the file if buffered may be 20 to 40 times larger than necessary. Such an expanded file should not be persisted or saved to disk, as other applications could introduce inconsistencies into duplicated entities. If all data were output this way, no referencing will be needed as every object will appear whenever needed. In this example the `IfcCartesianPoints` for a cube are repeated as needed.

```

<!--start of faces -->
<IfcFace>
  <Bounds>
    <IfcFaceOuterBound>
      <Bound>
        <IfcPolyLoop>
          <Polygon>
            <!--start of points -->
            <IfcCartesianPoint>
              <Coordinates>
                <IfcLengthMeasure>0.5</IfcLengthMeasure>
                <IfcLengthMeasure>0.5</IfcLengthMeasure>
                <IfcLengthMeasure>0.5</IfcLengthMeasure>
              </Coordinates>
            </IfcCartesianPoint>
            <!--and three other points -->

          </Polygon>
        </IfcPolyLoop>
      </Bound>
      <Orientation>true</Orientation>
    </IfcFaceOuterBound>
  </Bounds>
</IfcFace>
<!--and five other faces -->

```

4.6 Validation

4.6.1 XML

Most XML tools will report failure if the XML is not 'well-formed' due to incorrect headers, unbalanced quotes or angle brackets, mis-matched entity opening and closing etc.

Some web browsers including Microsoft's Internet Explorer will perform this checking.

4.6.2 ifcXML

XML workbenches such as XMLSPY or TurboXML will validate against the schema definition if available. NB TurboXML may generate warnings related to the schema that may be ignored.

4.6.3 IFC

Only by loading ifcXML documents into an EXPRESS model server can full validation against the original IAI IFC schema can be achieved.

5 Reading ifcXML

5.1 Starting

The parsing of an ifcXML file may choose to examine each entity in turn, or each entity at the top level. Filters can be applied and a set of entities matching certain criteria can be returned. If the intention is to traverse the whole model, then an application should first read the single [IfcProject](#), and then examine its attributes, which may include entities of interest, before examining the entities which refer to the [IfcProject](#) which typically will include aggregating relationships which should lead to the discovery of the [IfcSite\(s\)](#) or [IfcBuilding\(s\)](#) present. However ifcXML files can be used to transmit generic or detailed resources, such as [IfcMaterial\(s\)](#) or [IfcTypeProduct\(s\)](#) or their sub-types. In this situation a [IfcProject](#) entity should still be present, but may have no relationship to the remaining data.

For more information see section 2.5.6 of the ifc2x2/ifc2x3 model implementation guide

5.2 Alternation

A characteristic of the ifcXML schema is that when parsing down through the XML hierarchy, entities (with an [Ifc](#) prefix) will alternate with properties (without any such prefix). This remains true even when an internal or external link is followed, as long as the move from reference to target is executed but not counted. This hint may clarify browsing the model and simplify implementation when the model is being parsed systematically; for example an entity may use an 'href' but an attribute will not.

5.3 Href link

Applications can thrash up and down a large ifcXML file searching for the dependant entities. Every opportunity should be taken to reduce this overhead by generating search results once and re-using them. Internal indexes (look-up or hash tables) of 'href' and 'ref' references and their resolution should be used where available. This will aid efficient reading. An example using the 'key' functionality of XSLT is shown below.

5.4 Namespace

Depending on the exact header used, it may be legitimate to omit the namespace or use an arbitrary prefix. Applications should not rely on its presence or value.

5.5 Establishing units

If the application needs knowledge of the units of measurement in use, then the [IfcProject](#) should be examined and the 'UnitsInContext' attribute explored, as this attribute leads to a list of the default units assumed by the generating source. In particular the [IfcUnit](#) having 'UnitType' set to the enumeration value 'lengthunit' should be identified as all geometry resources will be in that unit. Other units may be listed to establish default units for quantity resources and default units for other properties. For more information on Units, see section 6 of the Ifc2x3 Model Implementation Guide.

5.6 Filtering

When working with utilities that support the XPATH filtering then care should be taken to anticipate the use of both XLINK and ordinary containment. Local (relative) filtering should be used wherever possible. This serves to avoid large searches and delays that can arise with global (absolute) paths.

6 Code samples

The following samples are offered for information without any warranty.

6.1 XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:fn="http://www.w3.org/2005/xpath-functions"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xmlns:ex=
    "urn:iso.org:standard:10303:part(28):version(2):xmlschema:common"
  xmlns=
    "http://www.iai-tech.org/ifcXML/IFC2x3/FINAL"
  xsi:schemaLocation=
    "http://www.iai-tech.org/ifcXML/IFC2x3/FINAL
    http://www.iai-tech.org/ifcXML/IFC2x3/FINAL/IFC2x3.xsd"
  >
  <xsl:output method="html" version="4.0"/>
  <xsl:variable name="bgcolor" >#cccccc</xsl:variable>
  <xsl:key name="href2id" match="*" use="@id"/>
  <xsl:template match="/" >
  <html>
    <head>
      <title>Schedule of Organizations</title>
    </head>
    <body bgcolor="{ $bgcolor }">
      <h1>Schedule of organizations</h1>
      <h4>
        <ul>
          <xsl:apply-templates
select="//*:IfcOrganization[not(@ref)]"/>
          </ul>
        </h4>
      </body>
    </html>
  </xsl:template>
  <!-- re-address any references onwards to the real things -->
  <xsl:template match="//*[@ref]" >
    <xsl:variable name="hvalue" select="@ref"/>
    <xsl:apply-templates select="key('href2id', $hvalue)"/>
  </xsl:template>
  <!-- report real things -->
  <xsl:template match="//*[not(@ref)]" >
    <li>
      <xsl:value-of select="name()"/>
      <xsl:if test="*:Name">
        <xsl:text> / name: </xsl:text>
        <xsl:value-of select="*:Name"/>
      </xsl:if>
    </li>
  </xsl:template>
</xsl:stylesheet>
```

6.2 `</xsl:stylesheet>`Java

Java coding depends on which XML toolkit is in use. This fragment is intended to show how having obtained an entity node, any 'ref' attribute can be resolved to the matching entity with an 'id' attribute.

```
// extract the xml identifier attribute of this node and jump any href link
to the id node
private Node resolveNode(Node pEntityNode) {
    Node aNode = pEntityNode;
    String xmlRef = aNode.getAttribute("ref");
    if (xmlRef.length() > 0) {
        String anXPath =
            "/" + aNode.getNodeName() + "[@id='" + xmlRef + "']";
        NodeList aList = XPathAPI.selectNodeList(aNode.getRoot(), anXPath);
        if (aList.getLength() > 0) {
            aNode = aList.item(0);
        }
    }
    return(aNode);
}
```

6.3 C++

(Readers are invited to submit sample code for inclusion here)

6.4 VB

(Readers are invited to submit sample code for inclusion here)

7 Questions and Answers

7.1 Why should I use ifcXML?

ifcXML offers a combination of advantages, first using XML technology as the mainstream technology for information publication and exchange, whilst reusing a well established, internationally recognized and supported data standard.

7.2 Do I have to learn and understand the whole IFC model ?

No, realistic exchange and data sharing scenarios particularly those that benefit most from ifcXML are based on subsets of the whole IFC model.

As an implementer of ifcXML you should have an overall understanding of what is in the IFC model in order to find the part that you are interested in. These subsets of IFC that fulfill a certain exchange case are named "views". So first look, if your part of interest is already described as an IFC view. If so learn and understand that subset of the IFC model that is described in the view. If a suitable view definition does not exist, than you should (together with the other parties interested in sharing the information) start to identify the scope for that view.

7.3 What is the best way to really understand the IFC model?

Start reading the IFC technical guide and the IFC model implementation guide, both give an insight into the IFC model from a technical and implementation view point. Currently both guides use the EXPRESS language version of the IFC model for examples and explanations.

7.4 What are IFC model views and how do I deal with views?

The IFC model scope covers the data needs for the whole life cycle of buildings and other facilities and for all disciplines involved over the life cycle (the current IFC2x3 model has reached a high degree of completeness compared with the overall scope of IFC). On the same time it is clear, that no single application will ever need to support all data for all disciplines over the complete life cycle.

The subsets of the IFC model that comply with specific exchange data needs at particular stages of a building life cycle phase are defined as "views", an example for a view is "quantity take-off from the architectural design model", or "takeover of the structural analysis model into structural design".

7.5 Does ifcXML replace SPF format?

No. For efficiency of large-scale data transfer within the AEC-FM domain, the SPF format will remain superior. As a rule of thumb the authors expect that the exchange and sharing of complete building information models (especially when including the full geometry representation) will remain mainly SPF format (*.ifc) based, whereas the exchange and reporting of partial models and reports (like a window schedule from the BIM) will strongly benefit from ifcXML. A model server can convert between the two formats if necessary.

7.6 Do I have to learn EXPRESS and SPF format to use ifcXML?

No, a detailed understanding of EXPRESS is not necessary in order to work with ifcXML, since the IFC specification is now also available as XML schema (the ifcXML schema). The same applies to SPF format, since ifcXML is exchanged as XML documents.

Currently the complete IFC model documentation (i.e. the explanatory text) is written around the source (i.e. EXPRESS) model, therefore when reading the documentation a base (not detailed) understanding of EXPRESS is beneficial. The IAI therefore has published a 10 page readers guide to EXPRESS reading this should be sufficient for reading the IFC model documentation.

For a brief understanding of the syntax of SPF format (*.ifc), the IAI has also published a readers guide to STEP physical file format.

7.7 Why is ifcXML more complicated than text book examples of XML?

ifcXML is an expression of the IAI IFC single unified schema for AEC-FM. Buildings are complex and information about buildings has a large number of inter-relationships between entities that mean models cannot always be represented as a single simple hierarchy. Furthermore, the trend is towards using combinations of schemas rather than just one. For ifcXML the 'ex' schema provides the overall wrapper for the data, and the 'xlink' schema provides extra functionality within the data. This sophistication is often not reflected in text book examples.

7.8 Can I send or receive parts of a whole building model?

The IFC model systematically separates resources such as materials, which are typically identified by name, from the building elements and relationships which are identified by the 'GlobalId'. An application may use these to aid the re-merging of any partial model.

In some cases it may be possible for the 'href' references to contain a full URL to a web based model server that can satisfy the referencing on demand.

7.9 Do I need an object model server to work with ifcXML models?

No, the ifcXML schema is an industry standard XSD. Many relational database applications and servers can read XSD as an alternative to SQL DDL and may be able to read and write ifcXML directly. However, many such implementations are only partial and a careful investigation should be made to identify the actual capabilities.

7.10 Can I store and retrieve ifcXML data using MS Access database?

This is possible using Access 2003, but it does not support (i.e. recognize) XLINK so most relationships are lost. Only data values are extracted into the tables.

- To load the 'structure' i.e. the schema

Use **File/Get_External_Data/Import** and select the ifcXML .xsd.

- To capture the data for use

Use **File/Get_External_Data/Import** and select an ifcXML .xml, loading into existing tables.

7.11 How does ifcXML relate to aecXML?

AecXML is a collaboration effort between many software vendors and dot.com companies that originate in North America to utilize XML in the AEC industry. After some delays and detours it is now committed to use ifcXML commercially and to augment ifcXML with other XML to enable business areas that are not covered by the IFC model. The intent is that these aecXML definitions will be submitted to the IAI for consideration in IFC model extensions.

More information at: <http://www.iai-na.org/aecxml/mission.php>

7.12 How does ifcXML relate to gbXML?

gbXML stands for "Green building" XML, an initiative to obtain basic building information to perform energy analysis. The gbXML initiative started in California, US. and by today is not internationally accepted nor endorsed as a standard. It represents only a small aspect of the full building information model and facility lifecycle, and is intended for early energy analysis based on concept designs.

More information at: <http://www.gbxml.org/>

Appendix 1 Earlier ifcXML methodologies

A 1.1 History

The IFC SPF format has been in use in the AEC-FM domain since 1996. In 2000, the BLIS organization proposed a methodology for representing EXPRESS based data in XML called BLIS-XML which was agreed to be the official XML representation for IFC 2.0 based data. In 2001 the International Technical Committee of the IAI commenced a project to agree an XML representation. The ifcXML-1 methodology was developed as a first attempt to generate an ifcXML schema directly from the IFC EXPRESS source definition, The IFC2x release was the first IFC version to which the ifcXML-1 methodology had been applied. The ifcXML-1 methodology is available at the IAI international website and was submitted to ISO TC184/SC4 as a requirement definition for developing part28 edition 2.

- www.iai-international.org/iai_international/Technical_Documents/ifcXML1.htm

A 1.2 Differences

This earlier ifcXML-1 was defined manually and differs from the current ifcXML in the following respects:

- Simpler header
- Ifc prefix on entities was dropped, the remainder being in CamelCase e.g. [BuildingStorey](#)
- Property names began in lower case e.g. elevation
- Co-ordinates, directions and date lists were expressed as a string. E.g. `<direction>1.0 0.0 0.0</direction>`
- 'globalId' was not constrained to be exactly 22 characters
- The root element was 'ifcXML'

A 1.3 Example (superseded)

```
<?xml version="1.0"?>
<ifcXML
  timeDeclarations="iso8601"
  xmlns="http://www.iai-international.org/ifcXML/ifc2x/IFC2X_FINAL"
  xmlns:xlink="http://www.w3.org/2001/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
    "http://www.iai-international.org/ifcXML/ifc2x/IFC2X_FINAL
    ifc2x_final_stage2.xsd">

<TypeProduct>
  <globalId>IP1000</globalId>
  <ownerHistory>
    <OwnerHistory href="oh1" xsi:nil="true"/>
  </ownerHistory>
  <name>IP1000</name>
  <description>Acme Hatch Trapezoidal IP1000 GRP</description>
  <representationMaps>
    <RepresentationMap>
      <mappingOrigin>
        <Axis2Placement2D>
          <location>
            <CartesianPoint>
              <coordinates>0. 0. 0.</coordinates>
            </CartesianPoint>
          </location>
        </Axis2Placement2D>
      </mappingOrigin>
    </RepresentationMap>
  </representationMaps>
</TypeProduct>
```

```

        </location>
        <refDirection>
            <Direction>
                <directionRatios>1. 0. 0.</directionRatios>
            </Direction>
        </refDirection>
    </Axis2Placement2D>
</mappingOrigin>
<mappedRepresentation>
    <ShapeRepresentation>
        <contextOfItems>
            <GeometricRepresentationContext href="BoundingBox"
                xsi:nil="true" />
        </contextOfItems>
        <representationIdentifier>Extent</representationIdentifier>
        <representationType>BoundingBox</representationType>
        <items>
            <BoundingBox>
                <corner>
                    <CartesianPoint>
                        <coordinates>-1000 -500 -47.5</coordinates>
                    </CartesianPoint>
                </corner>
                <xDim>2000</xDim>
                <yDim>1000</yDim>
                <zDim>95</zDim>
            </BoundingBox>
        </items>
    </ShapeRepresentation>
</mappedRepresentation>
</RepresentationMap>
</representationMaps>
<tag>Acme Hatch Trapezoidal IP1000 GRP</tag>
</TypeProduct>
</ifcXML>

```

Note: For brevity, the referenced [OwnerHistory](#) and [GeometricRepresentationContext](#) have been omitted.

A 1.4 Limitations

- Not compliant to Part 28 Edition 2
- Not directly mappable to or from SPF format (Part 21)
- Some entity names conflicted with SQL and other database reserved words and some entity names conflicted with attribute names. A suggested renaming was to use [lfc](#) as a prefix. This was only an issue with ifcXML-1 when used with some hybrid relational-object databases.

Approval	ConnectionGeometry	Constraint
Curve	Direction	Edge
Element	Face	FaceSurface
LibraryReference	Logical	Material
ObjectPlacement	OwnerHistory	Placement
Representation	WorkPlan	Vertex
Group	Space	Table
Zone		

A 1.5 Implementations (up until July 2007)

A.1.5.1 There are now no known implementations of ifcXML1.