IDM BCF Issue Management

Task Group BCF

Regional Chapter Stuttgart

Version: 1.0
Date: 30.09.2021

Contents

Abbreviations

MC	Model Checker
CDE	Common Data Environment
CAD	Computer Aided Design
FM	Facility Management

Acknowledgments

The following people and companies have contributed significantly to this publication:

Name	Funktion
Benedikt Fischer	BIM-Consultant, ComputerWorks
Dirk Holzmann	BIM-Manager, Drees & Sommer
Dorothee von Schnakenburg	BIM-Kompetenzzentrum Betriebsleitung, Vermögen und Bau Baden-Württemberg
Heiko Weigle	BIM-Koordinator, Weiske und Partner
Karina Idrisova	BIM-Kompetenzzentrum Betriebsleitung, Vermögen und Bau Baden-Württemberg
Lukas Martin	BIM-Koordinator, Weiske und Partner
Marcus Nill	BIM-Koordinator, Ingenieurgruppe Bauen
Patrick Koska	BIM-Manager, Vollack Gruppe
Peter Düchting	BIM-Manager, BAUWENS
Tobias Döring	BIM-Manager, hammeskrause architekten

Special thanks also to the vendors who provided us with copies of their software for testing purposes and the countless small contributions from various people not named directly.

Introduction

This paper is the result of 2 years of evaluation of the bcfXML 2.1 specification and its implementation within various software products by the German BCF task group of the regional chapter Stuttgart.

Since there has never been an open dialogue on user-defined requirements for the development of BCF before, the working group brought together as many end-users and software vendors as possible to map and test standard processes and identify the core parameters needed for day-to-day work.

Furthermore, the working group conducted real-life tests on software tools regarding the quality of implementation. While doing so it not only searched for bugs but also got a first-hand impression on how well the intended workflows and parameters suit real-life conditions.

During the whole process the working group gained deep insight into the status and quality of implementation and identified certain challenges which currently prevent industry-wide adoption of BCF workflows.

Herein the working group shares these findings as an impulse and reference for further development and future certification processes.

We are deeply convinced that bcfAPI will play a major role within the digital construction industry in the future and prioritized testing it whenever possible. Thus, all essential points mentioned in this document are equally applicable for file-based data exchange and bcfAPI.

Use-Cases

BCF was created for facilitating open communications and improving IFC-based openBIM processes by exchanging model-based issues between BIM software tools.

There are several use cases that benefit from BCF-enabled workflows, where information can be derived from the BIM and connected back to the BIM for object-specific information.

As a starting point we defined common use-case scenarios within a construction project test against:

1. Coordination of a clash found in a Model-Checker with 2 CAD-Systems
2. Client question in a CDE-System, answered in a CAD-System
3. Bilateral Communication between to CAD-Systems
4. Issue-Management on construction-site with a CDE-System

While testing these use-cases iteratively it became clear that they could be visualized in a generic process map.

Software

The following software tools were tested:

Program	Version	bcfApi/bcfXML
Allplan	2019/01	bcfXML
Bimplus	2019/06	bcfXML
ArchiCAD	22	bcfAPI/ bcfXML
BIMcollab Zoom	2019/01	bcfAPI/ bcfXML
Tekla BIMsight	1.9	bcfXML
California.pro	10.1.09	bcfXML
Nova AVA	2.5	bcfXML
Vectorworks	2020	bcfXML
DDS-CAD	16	bcfXML
Revit	2020.2	bcfAPI/ bcfXML
Solibri Office	9.9.5	bcfAPI/ bcfXML

* Summaries of the test results can be obtained from the working group.

Evaluation

Assessment of the actual implementation quality was done by analysis of imported and exported BCF files from widely used software tools. Solibri Office was used as reference implementation to test against. Whenever available bcfAPI was prioritized in testing. The test procedure is shortly explained in the following:
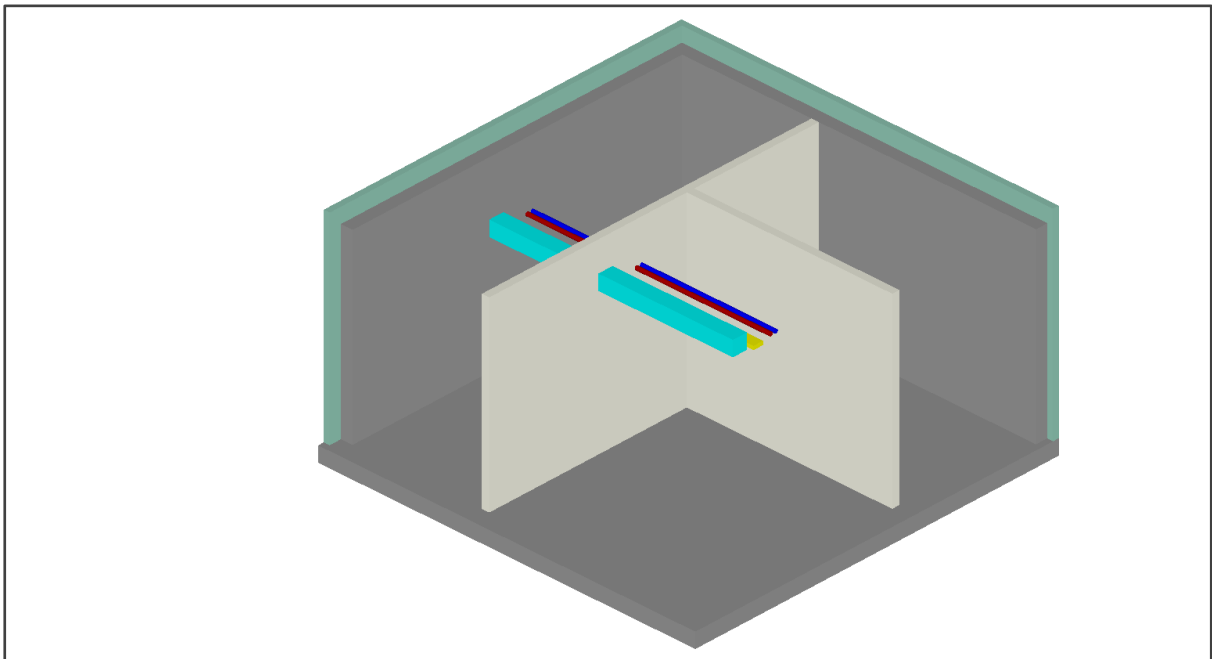
A: Creation of BCF test data

Figure 1 Screenshot of standardized IFC-Model

To make a meaningful comparison possible, all BCF files were created from a standardized IFC test model and a set of pre-defined attributes.

B: Import/Export test system

The content of each BCF file was then exported from Solibri Office and imported into the CAD program and back again. After this process we reviewed and compared against the bcfXML 2.1 specification, and the requirements identified within the use-case scenarios.
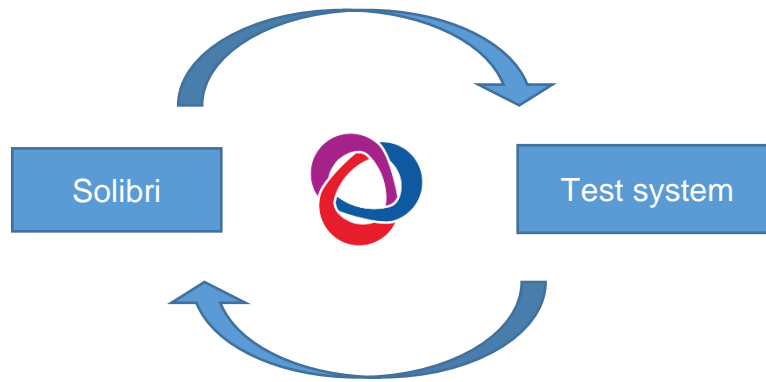
Figure 2 test schema

C: Evaluation

Figure 3 protocol sheet

The evaluation of the exported data from the test system was performed by means of a re-import into the reference implementation (Solibri Office) and an examination of the raw BCF data with an XML editor. All results have then been documented in a standardized sheet.

Conclusions

All in all, the results show a wide spread and no tool was able to export a fully compliant bcfXML 2.1 file. Furthermore, we found that no vendor has implemented BIMsnippets so far. Many also don't support attaching documents or links to BCF tickets, respectively.

These shortcomings can in most part be traced back to the lack of a proper certification process. The working group therefore highly recommends the establishment of a dedicated global testing and certification framework for BCF to improve and ensure consistent implementation quality in the future.

Additionally, the working group identified a set of crucial parameters that are missing in the current core schema. (see Appendix A) An expansion with respect to industry requirements on task management would increase acceptance and further propagate the use of BCF.

It also became apparent, that features such as rights management and full set of obligatory parameters are missing. As well as clear assignment to a person and identifiables such as a ticket number, labeling; categories and priorities should be expanded as well.

Editor's note: BCF 3 already addressed the issue of a missing human-readable ticket number.

BPMN

The use case describes four scenarios where two designers coordinate their issues with the help of a model checker.

The working group started out with 4 different process maps but soon realized, that these use cases can be visualized in one generic process.
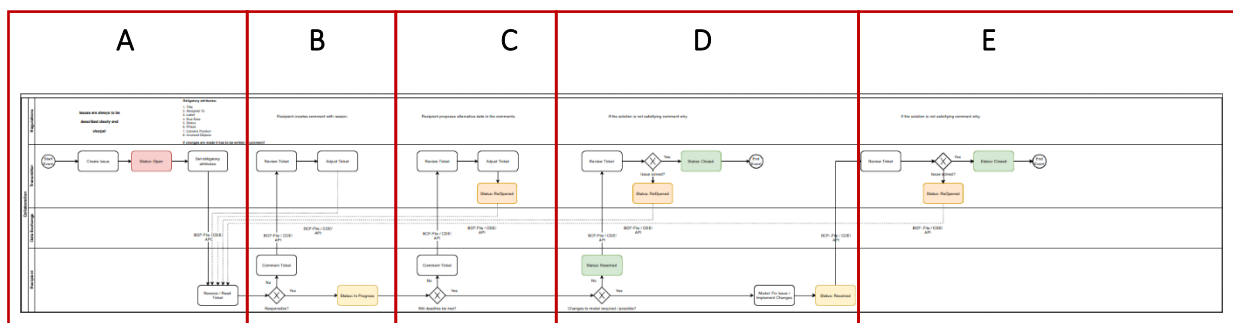
Abb. 1 See Appendix A for full map

A: Create ticket

The issue containing all the important details is created with status „Open". The author assigns the issue to the responsible person and sends it via file transfer or bcfAPI.

B: Receiver checks issue for responsibility

The receiver checks if he is responsible at all and comments accordingly if he is not.

After that the ticket is sent back to the sender or status is set to „In Progress".

- B.1: The original sender corrects the ticket according to the hints in the comment and resends it.

C: Receiver checks tickets due date

After reviewing responsibility and accepting the task a sanity check of the due date has to be performed. If the given time frame is not sufficient the issue has to be given back to the sender with suggestion of a new due date.

- C.2: The receiver suggests a new due date and sends back the ticket

D: Implement changes

The receiver implements the required changes and answers the ticket. Status is set to "Resolved" and the ticket is being sent back to the author.

E: Close ticket

The author performs a final check with regards to the points mentioned in the issue. Depending on the result of the check status is set to "Closed" or the ticket is reopened with Status "ReOpened" and sent to the receiver in another attempt to solve the problem.

Parameter

All parameters from the bcfXML 2.1 Standard have been thoroughly checked with regards to necessity and completeness.

Changes have been committed where legal aspects matter. Comments should be unchangeable, to ensure documentation. We also recommend logging timestamps and names of the involved IFC files.

Furthermore, we defined obligatory parameters, to ensure completeness of the issues.

Appendix A - Required parameters

Added / Changed Parameters are in red

name	valid entries	available in V2.1	obligatory	where to find	changeable by
Header					
GUID	UUID	X	X	Markup.bcf:Topic:Guid (folder name)	automatically
ProjectName	String	X	X	Project.bcfp:Topic:Name	automatically
InvolvedIfcFile(s) + Timestamp(s) ***	String Array	-			automatically
Topic					
TicketNumber	String	-	X	Human readable number for the ticket; alphanumeric	automatically
RuleSetName *	String	-			automatically
TopicTitle	String	X	X	Markup.bcf:Topic:Title	author
TopicType	Comment Issue Request Solution Task Clash	X	X	Markup.bcf:Topic:TopicType	author
TopicStatus	Open In Progress Resolved Closed ReOpened	X	X	Markup.bcf:Topic:TopicStatus	everybody except for Closed (author only)
Priority	high normal low	X	X	Markup.bcf:Topic:Priority	author
TopicLabel	Architecture Structure Mechanical Electrical Specifications Technology	X		Markup.bcf:Topic:Labels	everybody
CreationDate	Date	X	X	Markup.bcf:Topic:CreationDate	automatically
CreationAuthor	String	X	X	Markup.bcf:Topic:CreationAuthor	automatically
ModifiedDate	Date	X	X	Markup.bcf:Topic:ModifiedDate	automatically
ModifiedAuthor	String	X	X	Markup.bcf:Topic:ModifiedAuthor	automatically
DueDate	Date	X	X	Markup.bcf:Topic:DueDate	author
AssignedTo	String	X	X	Markup.bcf:Topic:AssignedTo	author
Description	String	X		Markup.bcf:Topic:Description	author
Continued on next page					

Stage (phase)	Preliminary Planning End Construction Start Construction End	X		Markup.bcf:Topic:Stage	everybody

DocumentReference					
DocumentReference (internal/external)	String	X		Markup.bcf:DocumentReference:ReferencedDocument	immutable, adding only

RelatedTopic					
RelatedTopic	UUID	X			everybody

Comment					
Comment	String	X		Markup.bcf:Comment:Comment	immutable, adding only
CommentAuthor *	String	X	X	Markup.bcf:Comment:Author	automatically
CommentDate *	Date	X	X	Markup.bcf:Comment:Date	automatically

Viewpoints					
Camera Position	Vector	X	X	Viewpoint_GUID.bcfv:Camera	everybody
Story *	String	-			automatically
Screenshot(s)	.png image file	X		Snapshot_GUID.png	immutable, adding only
Mark-Ups	Vector	X		Viewpoint_GUID.bcfv:Lines	everybody
Clipping Plane(s)	Vector	X		Viewpoint_GUID.bcfv:ClippingPlanes	everybody
InvisibleObjects	UUID	X		Viewpoint_GUID.bcfv:Components	automatically
InvolvedObjects	UUID	X		Viewpoint_GUID.bcfv:Components_Selection	automatically

* only if available/applicable
** CreationDate of IFC files