

Technical University of Denmark



Using BCF as a mediator for task management in building design

Treldal, Niels; Parsianfar, Hussain; Karlshøj, Jan

Published in:

Proceedings of the International RILEM Conference Materials, Systems and Structures in Civil Engineering 2016

Publication date:

2016

Document Version

Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):

Treldal, N., Parsianfar, H., & Karlshøj, J. (2016). Using BCF as a mediator for task management in building design. In Proceedings of the International RILEM Conference Materials, Systems and Structures in Civil Engineering 2016: BIM in Civil Engineering – Open Data Standards in Civil Engineering (pp. 48-59). Rilem publications.

DTU Library

Technical Information Center of Denmark

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

USING BCF AS A MEDIATOR FOR TASK MANAGEMENT IN BUILDING DESIGN

Niels Trelldal ⁽¹⁾⁽²⁾, Hussain Parsianfar ⁽²⁾, Jan Karlshøj ⁽¹⁾

(1) Technical University of Denmark, Lyngby, Denmark

(2) Rambøll Danmark A/S, Copenhagen S, Denmark

Abstract

buildingSMART has adopted the BIM Collaboration Format (BCF) to improve interoperability in the field of process information exchange. The original scope of BCF was linked to a need to communicate BIM-related tasks, but a further expansion of the BCF format should be considered to add additional support to industry requirements for task management. The research described in this paper was based on literature studies of industry needs and evaluations of the current BCF specification and its implementation in software, and it identified some challenges in the current state of BCF. Based on these findings, we propose an information system consisting of decentralised model and task servers using both BCF and IFC. Using IDM Part 2 as an example, we further propose an architecture to expand BCF.

1. Introduction

Building design is complex, not only because of complexity within one domain, but because of interdependencies between all three domains: product, process and organisation [1]. Building Information Modelling (BIM) [2] is a response to reducing product complexity while management approaches such as Lean Construction [3], Integrated Concurrent Engineering [4] and Agile Scrum [5] address process and organisational complexity. Task management, which is the focus of this research, is essential in such management approaches in combination with flow and value management [6]. Information exchange is crucial in all of these management approaches and buildingSMART addresses this need by providing standards to improve interoperability in the exchange of building information. From its beginning in the 1990s, buildingSMART focused primarily on supporting the product domain by developing the Industry Foundation Classes (IFC). The IFC specifications was intended to capture object-oriented building information and, over the years, the IFC specifications grew to include some process-related information. Today definitions for actors, time, approvals and action management are all included in the specifications [7].

1.1. The BIM Collaboration Format (BCF)

Combining product and process information in one data model generated a range of practical issues when exchanging this large data model as a bulk data transfer [8]. So in 2010, the software companies Tekla and Solibri came up with the proposal of an XML schema for the BCF format. The BCF format is an open file format that introduces workflow communication capability and can be connected to IFC models.

BCF contains information about a task (called a *topic* in BCF) including status, type and assignee, any comments related to the task, and references to related objects in an IFC model. It can also include information on the camera position/viewpoint location of the authoring tool as well as a snapshot of that view. In 2014, bcfXML v2 was released and adopted by buildingSMART [8]. This version included the option to append documents and elements of a data model (BIM snippets) as well as an option to include more viewpoints and snapshots [9]. The BCF format is independent of the IFC specifications, and enumerations for topic type, status, priority, etc. can be predefined by using an extension schema.

In real projects, hundreds if not thousands of BCF files are necessary for communicating tasks, and managing these as individual files can be difficult [10], [11]. To address this challenge, a specification for a BCF web service called bcfAPI v1 was developed along with the release of bcfXML v2. The specification defines a RESTful API, which allows a BCF server to automatically synchronise BCF tasks with others. Apart from an ability to define users more accurately (e-mail identification), the scopes of bcfAPI v1 and bcfXML v2 are identical, and the only difference is the way BCF tasks are exchanged. In the following, bcfXML v2 will be addressed in the comparisons carried out, but similar results would apply if bcfAPI v1 had been used as the communication format.

1.2. Process information spectrum

The original scope of BCF is linked to a need to communicate BIM-related tasks such as clash detection findings and other coordination issues. However, there are many other types of process-related activities in architectural, engineering and construction (AEC) projects, including the management of contract and user requirements, cost and risk management, interface coordination, site registration, etc. For this reason, buildingSMART and other organisations are also engaged in developing standards like IDM Part 2 [12] to define and manage contractual agreements and the Danish U106 Digital Defect Registration [13] to define and assign defects during construction. A review of these standards reveals that they share many aspects: they all address a specific task, include the creator and assignee, manage its status, and track the development of the task. Moreover, they also address the individual focus points such as BIM linkage in BCF, workflow definitions in IDM Part 2, and location information in U106.

Many tasks in construction are related, which results in the complexity referred to previously, but tasks can also evolve. For example, a clash might just be a task between two designers, but if the problem cannot be solved by changing the current design, the task might escalate to involve the client and even result in cost or time overruns affecting contractual issues. In this case, both BCF and IDM Part 2 are required to fully capture the process information involved. From an interoperability perspective, it is a challenge that multiple standards must be used in parallel because information might be lost in the conversion, and from an implementation

perspective, it is time consuming for software vendors if they are required to support multiple formats for similar functionality. In the light of these observations, it would be valuable to explore the possibility of harmonising standards related to capturing process information within management of tasks.

1.3. Study goals

The goals of this research were firstly to identify requirements for task management focusing on building design; secondly to assess the current implementation of BCF in available software to identify its current potential to capture process information; thirdly to compare BCF and IDM Part 2 to understand similarities and differences; and finally to evaluate the potential of expanding the BCF format to embrace a greater part of the spectrum of process information to support comprehensive task management.

2. Methodology

The research in this paper involved reviews of the literature and existing standards, and theoretical solutions are proposed on this basis. The research focused on building design, but most of the findings are applicable throughout the lifecycle of AEC projects.

2.1. Identifying requirements

To identify requirements for task management, the literature on current experience with AEC collaboration tools was reviewed and requirements specifically for task management were extracted. Findings in the literature selected were based on various approaches, such as interviews, implementation attempts, and research of existing tools. The findings are, therefore, believed to constituting a representative view of the industry needs including those not met by existing tools.

2.2. Review of bcfXML v2 exported by current software solutions

To review the content and quality of bcfXML information currently being shared, we made a structured review of the BCF packages exported from seven widely used tools supporting BCF. Each BCF package was created based on the same IFC model and a set of similar information was defined in each software solution to make it possible for their export capabilities to be properly compared. The content of each BCF package was reviewed in a text editor and compared to the bcfXML v2 specification. These findings were then compared to the industry requirements identified.

2.3. Comparison of BCF and IDM Part 2

To compare the BCF specification with the specification for IDM Part 2, we defined a list of the functionalities of the two standards and each specification was then evaluated based on these functional requirements. The purpose of this comparison was to provide a better picture of the areas where the two standards overlap and the areas where they differ.

2.4. Proposal for expanding the BCF specification

Finally, we used our findings and the findings from similar research to develop a theoretical proposal on how process and product information could be managed and exchanged within the framework of building design and how the BCF specification could be expanded to support this concept.

3. Requirements for a task management tool

The literature on systems to support the management of product, process and organisation complexity often defines such systems as collaboration tools. Table 1 identifies requirements in the literature that specifically target the task management part of collaboration tools. Requirements are grouped into categories of general requirements, requirements for what information is to be transferred, requirements for how workflows should be managed, and requirements for how tasks should be defined and related.

Table 1: Requirements for a task management tool

General	Information Transfer	Workflows	Task Management
<ul style="list-style-type: none"> • Improve understanding of project dependencies and impact of decisions [14], [15] • Support transparency [14], [16] • Provide overview, history, filtering, and any device accessibility [14], [15], [17], [18], [19] • Improve data management [14], [15], [20],[18], [17], [19] • Easy to use [14], [15], [20], • Support BIM model coordination [15], [20], [18], [17] • Include continuous system support and development [14], [20], [18], [17], [19] • Allow for integration with other tools such as simulation or visualisation tools [18] • Include social networking integration [14], [15] 	<ul style="list-style-type: none"> • Manage client decisions and client comments on project materials [14], [20] • Manage comments on design from other disciplines [14] • Document agreements and manage tasks agreed with the client [14] • Manage deviation reports and enquiries from the contractors [14] • Visualise interdisciplinary interfaces, and warn of potential negative impacts [14], [15], [17], [19] • Manage communication based on the BIM design at hand [14], [15], [20], [18], [19] • Manage interface coordination and level knowledge [14] • Managing what is not within the BIM model [14], [17], [19] 	<ul style="list-style-type: none"> • Manage user rights, roles and responsibilities [14], [20], [17], [19] • Provide guided and structured workflow [14], [15], [20], [19] • Allow for workflows to be adjusted [20] • Support for quality assurance [14] • Allow for agreeing on processes that support the needs of each discipline [14], [20] • Support automated workflows [14] 	<ul style="list-style-type: none"> • Ability to define categorisation of the tasks [14] • Ability to define prioritisation [14], [17] • Ability to relate to deadlines and phases [14], [15], [17] • Support better design scheduling [14], [15] • Manage responsibility for BIM information [14], [20], [17]

The review identified the importance in task management of capturing and saving the history of decisions, agreements, comments and questions. Tasks should be able to link to deadlines, phases and/or schedules, link to the BIM model, support data management, and link to other tools including social networking. The ability to refer to issues in documents or a specific point on the construction site is also desirable. Structured and yet adjustable workflows, user rights and roles must be supported, management activities should be automated, and it should be possible to categorise and prioritise tasks. The management tool itself should be able to visualise the information to improve understanding, leverage knowledge, and support transparency. The tool must also be easily accessible, easy to use, and be continuously supported and developed to support individual projects.

4. Evaluation of current BCF implementation

To assess the current implementation of BCF in tools that support task management, exported BCF XML files from seven selected tools were analysed and compared to the requirements in the bcfXML v2 specification [21]. The following tools were selected: BCFier, KUBUS BIMCollab, Trimble Connect, Solibri Model Checker, DDS Viewer, BIMTrack and Revizto. A summary of the results is shown in Table 2.

Table 2: Summary of results from bcfXML v2 exports – displaying missing, incorrect, supported and not supported content

Software (version)	BCFier (2.0.2.0)	BIMCollab (2.5)	Trimble Connect*	Solibri (9.6.12)	DDS Viewer (1.2)	BIMTrack (1.3)	Revizo (4.1.35834)
<i>File: bcf.version</i>	Not exported	Not exported	Not exported	Not exported	Inconsistent implementation: GUID not used	Not exported	Not exported
<i>File: project.bcfp</i>	Not supported	Not supported	Optional but not supported: IfcSpatialStructureElement.IsExternal, Filename, Date, Reference	Optional but not supported: IfcSpatialStructureElement.IsExternal, Filename, Date, Reference	Not supported	Not supported	Not supported
<i>File: mar-kup.bcf</i>	Optional but not supported: IfcProject, IfcSpatialStructureElement.IsExternal, Filename, Reference	Optional but not supported: AssignedTo Inconsistent implementation: Title - Description is placed in Title Labels - IssuedID in tool placed in Labels	Optional but not supported: Index, ReferenceLink, Labels	Optional but not supported: ReferenceLink, Priority, Labels	Mandatory but not supported: CreationDate Optional but not supported: Priority, Labels, ModifiedAuthor	Mandatory but not supported: CreationDate Optional but not supported: ReferenceLink, Labels, ModifiedDate, AssignedTo	Optional but not supported: Topology, ReferenceLink, Priority, Index, Labels, Modified/Author
<i>Header</i>	Optional but not supported: IfcProject, IfcSpatialStructureElement.IsExternal, Filename, Reference	Optional but not supported: AssignedTo Inconsistent implementation: Title - Description is placed in Title Labels - IssuedID in tool placed in Labels	Optional but not supported: Index, ReferenceLink, Labels	Optional but not supported: ReferenceLink, Priority, Labels	Mandatory but not supported: CreationDate Optional but not supported: Priority, Labels, ModifiedAuthor	Mandatory but not supported: CreationDate Optional but not supported: ReferenceLink, Labels, ModifiedDate, AssignedTo	Optional but not supported: Topology, ReferenceLink, Priority, Index, Labels, Modified/Author
<i>Topic</i>	Mandatory but not supported: CreationDate, CreationAuthor Optional but not supported: TopicType, TopicStatus, ReferenceLink, Priority, Index, Labels, ModifiedDate, ModifiedAuthor, AssignedTo	Optional but not supported: AssignedTo Inconsistent implementation: Title - Description is placed in Title Labels - IssuedID in tool placed in Labels	Optional but not supported: Index, ReferenceLink, Labels	Optional but not supported: ReferenceLink, Priority, Labels	Mandatory but not supported: CreationDate Optional but not supported: Priority, Labels, ModifiedAuthor	Mandatory but not supported: CreationDate Optional but not supported: ReferenceLink, Labels, ModifiedDate, AssignedTo	Optional but not supported: Topology, ReferenceLink, Priority, Index, Labels, Modified/Author
<i>BIMsnippet</i>	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
<i>DocumentReference</i>	Not supported	Not supported	Not supported	Not supported	Not supported	Optional but not supported: IsExternal	Not supported
<i>RelatedTopic</i>	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
<i>Comment</i>	Optional but not supported: ReplyToComment, ModifiedDate, ModifiedAuthor	Mandatory but not supported: Status Optional but not supported: VerbalStatus, ReplyToComment, ModifiedDate, ModifiedAuthor	Mandatory but not supported: Status Optional but not supported: VerbalStatus, Viewpoint, ReplyToComment, ModifiedDate, ModifiedAuthor	Mandatory but not supported: Topic Optional but not supported: ReplyToComment, ModifiedDate, ModifiedAuthor	Optional but not supported: ReplyToComment, ModifiedDate, ModifiedAuthor Inconsistent implementation: Viewpoint – Comments, reference under viewpoints, should be other way around	Mandatory but not supported: Status Optional but not supported: VerbalStatus, Viewpoint, ReplyToComment, ModifiedDate	Optional but not supported: Status Optional but not supported: VerbalStatus, Viewpoint, ReplyToComment, ModifiedDate
<i>Viewpoints</i>	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
<i>File: Visualization information (.bcfv)</i>	Not supported	Not supported	Optional but not supported: Color, OriginationsSystem, AuthoringToolId	Optional but not supported: Color	Optional but not supported: Color, AuthoringToolId	Optional but not supported: Color, AuthoringToolId	Optional but not supported: Color, AuthoringToolId
<i>Components</i>	Optional but not supported: Color Inconsistent implementation: Selected - Selected objects are not set to Selected=True, Visible - Used only when Visible = False	Optional but not supported: Color, OriginationsSystem, AuthoringToolId Inconsistent implementation: Selected - Used only when Visible = True, Visible - Hidden objects are not set to Visible = False	Optional but not supported: Color, OriginationsSystem, AuthoringToolId	Optional but not supported: Color	Optional but not supported: Color, AuthoringToolId Inconsistent implementation: Selected - Selected objects is not set to Selected = True, Visible - Hidden objects are not set to Visible = False	Optional but not supported: Color, AuthoringToolId Inconsistent implementation: Selected - Selected objects is not set to Selected = True, Visible - Hidden objects are not set to Visible = False	Optional but not supported: Color, AuthoringToolId Inconsistent implementation: Selected - Selected objects is not set to Selected = True, Visible - Hidden objects are not set to Visible = False
<i>OrthogonalCamera</i>	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
<i>PerspectiveCamera</i>	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
<i>Lines</i>	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
<i>ClippingPlanes</i>	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
<i>Bitmap</i>	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported



One or more required elements and/or attributes missing
 Inconsistent use of one or more elements and/or attributes
 Elements and attributes supported in accordance with bcfXML v2
 One or more optional elements and attributes not supported
 * Version not available. Latest BCF export from the website: 28-06-2016

The review in Table 2 shows the rather inconsistent comparison of the specification of the tools selected. No tool is able to export a BCF file fully compliant with the current 2012 specification, and elements and attributes available in the tools are not always exported.

Lack of implementation of non-optional attributes is seen in relation to *Topic*, where *CreationDate* and *CreationAuthor* are not exported in some cases, and in relation to *Comment*, where *Status* and related *Topic* are not supported in some cases. Incorrect implementation of attributes is seen in relation to *Components* where there is limited consistency in the methods of defining selected and visible objects, and in various other areas where attributes are not defined in accordance with the specification. Limited implementation of optional attributes is seen in relation to the *Header*, *BIMSnippet*, *DocumentReference*, *RelatedTopic*, *Lines*, and *Bitmap*. In several cases, the limited implementation is due to a lack of support for such functionality in the tools, but in many cases the functionality is supported in the tools but is still not exported. For example, viewpoints are not exported from Trimble Connect or Revizto although these are defined in the tools, and components are not defined from BIMTrack or Revizto even when selected here. The inconsistent implementation generates uncertainty in information exchange options because the different tools interpret BCF files differently. In almost all cases, round-tripping a BCF file through more than one tool will also result in loss of information due to the inconsistent implementation.

4.1. Support of industry needs by BCF

If we compare these findings with the industry needs in Table 1, most of the current implementations support a range of the general needs, such as information capturing of decisions, comments, and linkages to BIM design. However, more consistent implementation will be required to comply with needs such as assignment to a person (workflows), categorisation, prioritisation, labelling, relationships and document-linking (data management). The need for integration with other tools is partially supported by allowing for system-specific identification and open information exchange. Linking to deadlines, phases and/or schedules and referring to issues in documents or a specific location on the construction site are currently not supported. Similarly, the need to manage workflows, including user rights, roles and responsibilities, is only partially met in the current specifications. In the light of these findings, the BCF specification needs to be expanded if it is to fully support the needs of the industry. As an alternative, the functionality required could be implemented in task management tools, but interoperability in process information exchange would remain limited. For this reason, the following sections will elaborate on the potential for expanding the BCF format.

5. Comparison of BCF and IDM Part 2

IDM Part 2 specifies a methodology and XML format for describing coordination actions between actors in the construction industry [12] and is currently being used most intensively in the Netherlands under the name VISI to manage contractual agreements [22]. It has similarities to BCF, but IDM Part 2 also addresses predefined workflows, conditions, user rights, roles and organisations – features missing from the BCF format as described above. IDM Part 2 uses an extension schema to include an ability to predefine transactions and the messages required to support them. The extension schema can also be used to predefine most other attributes of the specification, and the intention therefore is to define process information more rigidly than was intended for BCF.

Table 3 compares the specifications of the two standards and identifies similarities and differences in BCF and IDM Part 2. The comparison shows that the two standards have many similarities, with workflow support in IDM Part 2 and BIM support in BCF being the main differences. However, neither standard is superior to the other when their abilities are compared with the requirements identified in Table 1. The requirements list both predefined workflows and BIM support as needed, along with a number of the elements and attributes that are present in one or other of the two standards. This suggests it might be best to try for a harmonisation of the two standards rather than selecting either one.

IDM Part 2 is based on organising transactions and messages in predefined or at least hierarchical order. This can be valuable in documenting agreements, but it runs the risk of making everyday tasks overly complicated to define. In this light, it is better to use the BCF specification as the starting point for a harmonisation, and to use the principles from IDM Part 2 to add additional methodology and attributes.

When parts of IDM Part 2 are implemented, the hierarchical relationships required by the specification should be avoided because they do not match the needs in other design activities. Instead, the focus should be on adding missing attributes and support for defining hierarchical relationships on top of the crosswise relationships.

6. Proposal for an information system for building design

To meet the information exchange needs to support task management, we have defined a proposal for an information system that captures and exchanges both product and process information. Centralising the management of process information on a task management server has proven valuable in relation to providing a quick overview on the status of a project and allowing for continuous commenting [11]. The idea of using a task management server is aligned with the previously described reasons for developing the bcfAPI web service to avoid

Functionality	bcfXML v2	IDM Part 2: 2012
Identification	GUIDs	Custom IDs
Extension schemas	Only enumerations for topic type, topic status, topic label, priority, users and snippet types	Extensive schema for entire specification incl. enumerations, workflows, user and roles, conditions etc.
Ease of use	Simple XML structure	More complex XML structure with several relationships required
User Management	Managed only by names	Unique users can be linked to roles and organisations
Topic Definition	ID, Title, Description, References, Priority, Labels, Index	Similar to BCF, but no priority or labels can be assigned. Topics can be predefined as required transactions or messages.
Workflow	Only a assigned user can be defined	Can be predefined with fixed workflows, statuses, conditions, roles and responsibilities
Task relationships	Simple crosswise 1:1 relationships	Hierarchical relationships but not crosswise relationships
Status	Simple status and date stamp	Status, state, send and received log and date stamp
Comments and notifications	Only comments supported	Comments supported as elements, notification requirements only as a Boolean value
Document support	Support for documents management internally or externally.	Support for documents management externally. Richer metadata compared to BCF.
Data exchange support	With snippets parts of any structured data can be exchanged	Only document support
BIM and viewpoint support	Direct linkage to IFC models and IFC objects and support for viewpoints	Not supported

BCF file exchange. However, researchers have also concluded [11] that it is a challenge to manage confidential information from different companies on a central server and have suggested a decentralised approach to overcome this problem. In recent years, a similar challenge has been discussed in the field of management of product information on model servers. Here, research has shown that a decentralised solution is more appropriate with model servers used only as reference models to one another [10], [23]. The conclusion is that there is only a limited need to edit work originating from other parties due to an already sharp split of responsibility. Harmonising the range of IFC interfaces in the different software tools to allow for error-free interoperability during editing and round-tripping of product information therefore still seems too resource-intensive to be attractive to the AEC industry.

Managing process information differs to some extent from managing product information because different parties need to be involved in process-related activities across different platforms as identified in Table 1. Specific users will need access to more than one server to manage different types of tasks – often using individual tools for each specific activity. Workflows and user rights on each platform or server might restrict different users in what they are allowed to do, but users from different domains need to be engaged in most activities. However, securing full interoperability between the different types of process activities and task servers seems of limited value as long as information is available for reference, e.g. to generate an overview of the project status or to generate new tasks based on existing ones. This is similar to the setup for model servers promoting the need for a decentralised server setup also for task servers.

In such a setup, the tools to manage process activities could be directly focused on carrying out a specific process activity and complying with the requirements in Table 1 of being easy to use and including guided workflows and leverage knowledge. The tools could use BCF to communicate with their dedicated server and, where a tool needs information on tasks from others servers, they could use BCF to query other servers for the information required.

A proposal for a setup with decentralised model servers and task servers is illustrated in Figure 1.

Each of the process activities is handled by one or more tools communicating with one or more task servers. Each task server can contain its own user management setup, workflow definitions and document storage to function independently. Preferably, the bcfAPI will be used to exchange information with the different tools and link the different servers to act as a coherent information system.

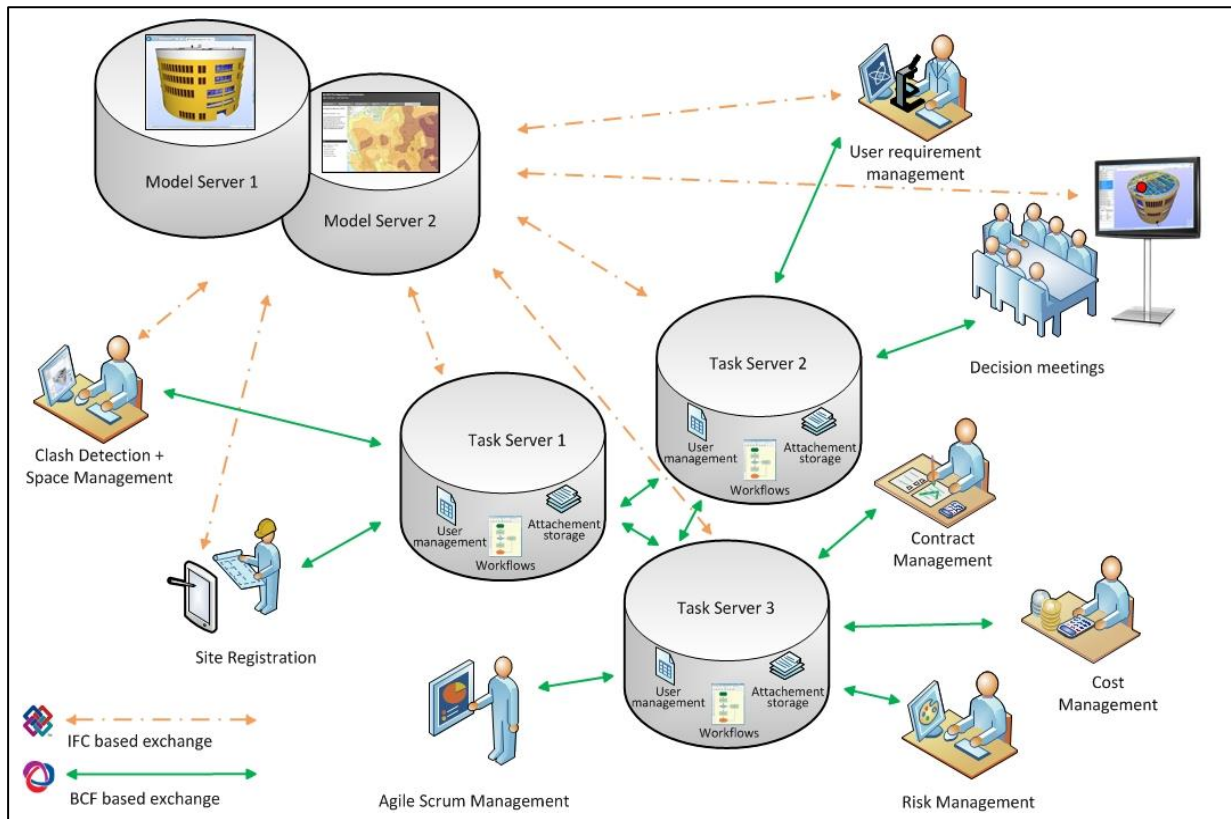


Figure 1: Proposal for information management and exchange in building design

Product information is collected, preferably using IFC exchange, from one or more model servers, as already proposed elsewhere [10], [23]. If required by the client or others, predefined workflows, required users and other conditions could be exchanged by implementing elements from IDM Part 2 in BCF to ensure consistent workflows across different platforms.

Should a project manager need e.g. to evaluate the overall project status or investigate why a certain area of a building or type of building component is causing problems, he or she will use a tool that, either by itself or via a task server, queries the other task servers to collect the task information required. To support such a setup, it would be preferable if the BCF specification used URIs (e.g. <https://server.com/bcf/projects/F445F4F2-4D02-4B2A-B612-5E456BEF9137/topics/B345F4F2-3A04-B43B-A713-5E456BEF8228>) instead of just GUIDs (e.g. B345F4F2-3A04-B43B-A713-5E456BEF8228) to identify not only the unique tasks but also the location of the task. This would ensure that any system would know exactly where the original task is located and should be updated, if required. It would also be beneficial if bcfAPI could be expanded to include standardised query calls, e.g. to present every task on a server that addresses a specific IFC object or contains a specific word in its title or description.

Practical implementation will be required to further evaluate how well BCF supports the proposed setup, but we did not identify any other aspects of the current BCF specification that would require adjustments for the setup to be supported.

7. Potential expansion of the BCF specification

One main advantage of the BCF format is its simple structure, which makes it easy to understand and implement [10], [11]. Expanding the scope of the BCF specification to include contract management methodology from IDM Part 2 or support of other process-related activities will make the specification more complex and potentially limit some of its current momentum in the industry at both implementer and practitioner level. The IFC specification was originally challenged by the same need to expand, so a layer-architecture was defined that provides the data schema with a modular structure to ease future development and allow for implementation to be selective and reusable [24].

The data schema structure has a resource layer at the bottom, a core layer in the middle, and an interoperability layer and a domain layer on top [7]. Fundamental structures and classes are defined in the core layer, and shared classes are defined in the interoperability layer. Domain-specific classes are defined in the domain layer, and resource definitions used in other layers are defined in the resource layer. The architecture is built on a “ladder principle” [24]. At any layer, a class may reference a class in the same or a lower layer, but may not reference a class from a higher layer. This allows for software implementation to be selective, because implementation of a class (in most cases) requires only implementation of the class and classes in lower layers.

If the BCF format is expanded, a similar architecture could be applied to achieve similar benefits. In the light of the findings in this paper, a proposal for a BCF data schema architecture was developed and is illustrated in Figure 2. There is a core part of BCF that is needed to support the requirements identified and the use cases illustrated. This includes definition of the topic, users involved, dates, commenting, simple relationships and, to support the requirements of IDM Part 2, hierarchical relationships. Together, these constitute the core layer. In the interoperability layer, linkage to documents, BIM and data snippets are defined because these are applicable to many activities. The domain layer from IFC is here called the activity layer because it represents definitions for specific process-related activities, such as contract, cost or risk management. For example, the activity schema for cost

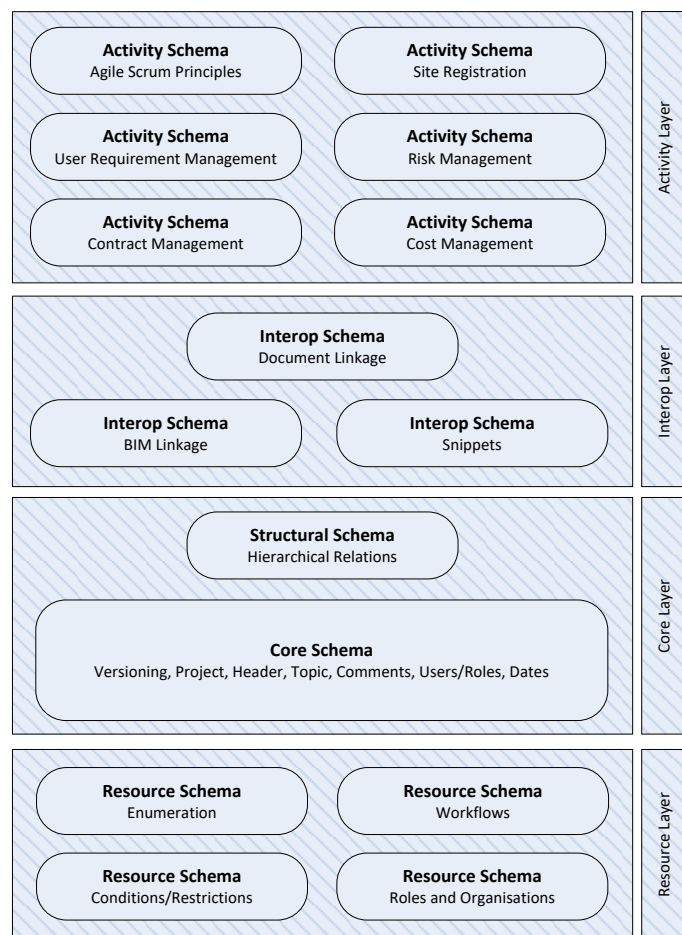


Figure 2: Proposal for an expanded BCF data schema architecture

management could include attributes for the cost of services or contracts, and for contract management the schema could include attributes for the send and receive log.

In the resource layer, expansions to incorporate several of the additional options of IDM Part 2 could be included because these can act as support for several activities. The resources could include options to define workflows, conditions, roles and enumerations for selected attributes. The six activity schemas are examples of activities that could be implemented, but actual industry needs should define the priorities for implementation. In the light of the findings in section 4, implementing missing elements from IDM Part 2 would in itself address most of the current needs of the industry.

At the same time, moderation in the range of activities supported should be considered to keep the specification simple. The existing functionality of BIM Snippets, which provides an option to exchange parts of a different data structure, could be used not only to exchange BIM data, but any relevant data. For example, a workflow defined in the BPMN format [25] could be attached as a *Snippet*, instead of requiring an extension of BCF to define complex workflows. In the light of the literature reviewed in this paper, there seems to be a difference between when an activity requires only one or more attributes to be fully supported by BCF and when it requires significant expansions to BCF to be supported. In the latter case, the use of snippets should be considered. Implementing workflows in BCF could therefore be done by defining a resource schema for simple transactional workflows in BCF and promoting an implementers' agreement to use BPMN to support more advanced workflows. BPMN shares significantly fewer similarities with BCF than IDM Part 2, which makes harmonisation less attractive.

8. Conclusions

Expanding the BCF specification will allow BCF to support the exchange of information for a broader range of process activities and add considerable value in an open and decentralised information system for the AEC industry. Using the IDM Part 2 specification as a starting point will support several key needs of the industry. The current BCF specification has limitations and the current implementation is error-prone. Along with an expansion of the specification, a certification solution similar to the official IFC certification should be considered. As with the IFC certification, a BCF certification could start by focusing on support for elements and attributes in the proposed core and interoperability layers. Practical testing of the proposed setup and BCF specification architecture will be required to determine the desired rate and direction of an expansion.

9. Acknowledgements

We would like to thank Léon van Berlo from TNO, Klaus Linhard and Georg Dangl from iabi, and Ruben Blair from Bakker & Spees for sharing their knowledge on BCF and IDM Part 2 and engaging in valuable discussion on the research topic.

References

- [1] E. Pikas, L. Koskela, M. Thalfeldt, B. Dave, and J. Kurnitski, "Complexity in Designing Energy Efficient Buildings: Towards Understanding Decision Networks," in *Proceedings of IGLC23 – 23rd Annual Conference of the International Group for Lean Construction* (2015) pp. 213–222.
- [2] C. Eastman, P. Teicholz, R. Sacks, and K. Liston, *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, 2nd Ed. (2011)

- [3] L. Koskela, "An Exploration towards a Production Theory and its Application to Construction," *VTT Publications* (2000)
- [4] J. M. Chachere, J. Kunz, and R. Levitt, "The Role of Reduced Latency in Integrated Concurrent Engineering," *CIFE Working Paper*, no. 116, April (2009)
- [5] J. Sutherland, *The art of doing twice the work in half the time*. New York, US: Crown Business, 2014.
- [6] L. Koskela, P. Huovila, J. Leinonen, "Design Management In Building Construction : From Theory To Practice," *J. Constr. Res.*, vol. 03, no. 01 (2002), 1–16
- [7] BuildingSMART, "Industry Foundation Classes - Version 4 - Addendum 1," (2015). [Online]. Available: <http://www.buildingsmart-tech.org/ifc/IFC4/Add1/html/>. [Accessed: 19-Jul-2016].
- [8] BuildingSMART, "BCF intro," (2016) [Online]. Available: <http://www.buildingsmart-tech.org/specifications/bcf-releases/bcf-intro>. [Accessed: 19-Jul-2016].
- [9] BCF Implementer Support Group, "XML specification for BIM Collaboration Format" (2016) [Online]. Available: <https://github.com/BuildingSMART/BCF-XML>. [Accessed: 19-Jul-2016].
- [10] K. Lindhard and R. Steinmann, "BIM-collaboration processes - from fuzziness to practical implementation," in *European Conferences On Product And Process Modeling In The Building Industry (ECPPM)* (2014), 27–32
- [11] L. van Berlo and T. Krijnen, "Using the BIM Collaboration Format in a Server Based Workflow," *Procedia Environ. Sci.*, vol. 22 (2014), 325–332
- [12] ISO 29481-2, *Building information models - Information delivery manual - Part 2: Interaction framework*, Geneva (2012)
- [13] *U106, Digital Mangelinformation (in Danish)*. Herlev, Denmark (2014)
- [14] M. H. Parsianfar, "Open formats in a communication system," Technical University of Denmark, Kgs. Lyngby (2016)
- [15] S.-A. Kim, Y. Choe, M. Jang, and W. Seol, "Design process visualization system intergrating BIM data and performance-oriented design information," *Proc. 28th Int. Symp. Autom. Robot. Constr.* (2011), 728–733
- [16] S. Bertelsen, "Construction as a complex system," *Proc. IGLC*, vol. 11 (2003), 143–168
- [17] M. T. Shafiq, J. Matthews, and S. R. Lockley, "A study of BIM collaboration requirements and available features in existing model collaboration systems," *J. Inf. Technol. Constr.*, vol. 18 (2013), 148–161
- [18] P. Coates, Y. Arayici, L. Koskela, M. Kagioglou, et al. "The limitations of BIM in the architectural process," *First Int. Conf. Sustain. Urban. (ICSU 2010)*, no. December (2009), 9
- [19] M.T. Shafiq, J. Matthews and S. Lockley, "Requirements for model server enabled collaborating on building information models," *First UK Acad. Conf. on BIM, 5–7 Sept. 2012, Northumbria Univ. Newcastle upon Tyne, UK* (2012)
- [20] S. G. Yeomans, N. M. Bouchlaghem and A. El-Hamalawi, "Provisions for proficient Construction Project Extranet Protocols to facilitate Collaborative Extranet Working," *Proc. 22nd CIB-W78 Conf. Inf. Technol. Constr. Dresden, Ger.* (2005), 155–162
- [21] BuildingSMART, "BIM Collaboration Format v2.0 Technical Documentation," *GitHub* (2015). [Online]. Available: <https://github.com/BuildingSMART/BCF-XML/tree/master/Documentation>. [Accessed: 22-Jun-2016].
- [22] J. G. Hoeber, D. M. Alsem, P. H. Willems, "The management of information over the life-cycle of a construction project using open-standard BIM," *Proc. of the 32nd CIB W78 Conference, Eindhoven, The Netherlands* (2015), 295–301
- [23] L. A. H. M. van Berlo, J. Beetz, P. Bos, H. Hendriks, and R. C. J. van Tongeren, "Collaborative engineering with IFC : new insights and technology," in *European Conferences On Product And Process Modeling In The Building Industry (ECPPM)* (2012), 811–818.
- [24] IAI, "IFC Object Model Guide," International Alliance of Interoperability (1999)
- [25] OMG, "Business Process Model and Notation (BPMN) Version 2.0," Needham, US (2011)