

This project is funded by the Horizon 2020 Framework Program of the European Union

Built2Spec

Built to Specifications – Tools for the 21st Century Construction Site
H2020 Grant Agreement – 637221

D4.5 Use of BIM technology for self-inspection and quality checks

Primary Author: Léon van Berlo (TNO)

Contributors: VRM, PHI

1st Quality reviewers: Marcus Keane and Magdalena Hajdukiewicz (NUIG)

2nd Quality reviewer: Ian Wallis (BSRIA)

Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	M30 – July 31 st , 2017
Actual delivery date:	27 June 2017
Version:	1.0
Total number of pages:	50
Keywords:	BCF, BIM, building industry, IFC, self-inspection, quality check

DISCLAIMER

The opinion stated in this report reflects the opinion of the authors and not necessarily the opinion of the European Commission. Neither the EACI nor the European Commission are responsible for any use that may be made of the information contained therein.

All intellectual property rights are owned by the BUILT2SPEC consortium members and are protected by the applicable laws. Unless otherwise specified, all document contents are '@BUILT2SPEC – All rights reserved'. Reproduction is not authorized without prior written agreement.

The commercial use of any information contained in this document may require a license from the owner of that information.

All BUILT2SPEC consortium members are committed to publish accurate and up to date information and take the greatest care to do so. However, neither the European Commission nor the BUILT2SPEC consortium members can accept liability for any inaccuracies or omissions nor do they accept liability for any direct, indirect, special, consequential or other losses or damages of any kind arising from the use of this information.

ACKNOWLEDGEMENT

This document is a deliverable of the BUILT2SPEC project which has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement no. 637221.

Document History

Version No.	Date	Author	Revision Description
0.1	26/02/2016	TNO	Initial draft
0.2	09/01/2017	TNO	Update from internal review
0.3	10/03/2017	TNO	Update based on new integration with VCMP
0.4	26/04/2017	TNO	Integrated subset of publications
0.5	09/06/2017	TNO	First draft shared with project team
0.6	10/06/2017	ETH & VRM	Integrated contributions from ETH/VRM
0.7	10/06/2017	TNO	Ready for review
0.8	26/06/2017	TNO/BSRIA/NUIG	Processed reviewers comments
1.0	27/06/2017	TNO	Final version

Executive summary

‘Built2Spec - Tools for the 21st century construction site’ project brings together a new and breakthrough set of technological advances for self-inspection and quality assurance that will be put into the hands of construction stakeholders to help meet Europe’s energy efficiency targets, standards for constructing and retrofitting buildings, and related policy ambitions.

The centrepiece of the Built2Spec project is a cloud based virtual construction management platform (VCMP), conceived following the most advanced integrated design and delivery framework for the building sector. The VCMP will host applications that facilitate worksite activities and quality compliance by putting knowledge in hands of contractors. This will be done through shared design specifications and 3D models, installation guidelines, information on regulatory frameworks, and help from construction experts on smartphones and tablets, from initial design to delivery.

Within WP2 and WP3 project partners develop advanced techniques for Thermal Imaging, Air Tightness, Indoor Air Quality and Acoustic testing. These tests may use input from a BIM dataset to perform measurements, and should be able to link results to objects in a BIM dataset.

BIM stands for ‘Building Information Management’ and is about modelling a building in terms of data (object, relations, properties, etc.). With the uptake of BIM the building industry is changing from a paper-driven industry to a data-driven industry. **The goal of Built2Spec task 4.5 and the content of this report is “to make BIM technologically ready for self-inspection and quality checks”.**

This report is a direct input to WP5 where the overall objective is to develop the detailed process steps for self-inspection and quality checks that a site worker and site supervisor would follow to acquire the data needed to automatically generate project files and reports which demonstrate compliance with the relevant standards, specifications and regulations.

This report concludes that the proposal from D1.4 to use linked services is the right approach. This report suggests to use BIM Bots for WP5.

This report also concludes that using open BIM data standard IFC is recommended for BIM data exchange. Modelling requirements need to be respected to ensure necessary data is available for each self-inspection and quality check. There is still no standard BIM Query language to recommend in this project. The GUID integrity issues when using IFC can be ignored using an appropriate workflow. Issue management and results from self-inspection and quality checks is best communicated using the BIM Collaboration Format (BCF). Using self-learning algorithms to predict and prevent constructability issues on site, is a promising technology. Using IFC/BIM Data online can be facilitated by an online WebGL viewer (BIM Surfer).

Table of Contents

Document History	3
Executive summary	4
Table of Contents.....	5
Abbreviations.....	7
1 Introduction	8
1.1 The Built2Spec project	8
1.2 Goal and context within the project.....	8
1.3 Research topics	9
2 BIM.....	10
2.1 Software Tools	10
2.2 Misconceptions.....	11
2.3 BIM Data Standards	12
2.4 BIM Web Service Standards.....	13
2.5 BIM Issue Management	13
3 BIM integration	15
3.1 Distributed approach	15
3.2 Integration of different systems	16
4 Generic use case Built2Spec	17
4.1 VCOMP generic process flow.....	17
4.2 Tools.....	17
4.3 BIM technology and data flow.....	19
4.4 Example.....	20
4.5 Reading guide for next chapters.....	21
5 Requirements to BIM data for quality checking on-site.....	23
5.1 BIM Standards.....	23
5.1.1 Classifications.....	24
5.1.2 Model View Definitions.....	24
5.2 General BIM Data requirements.....	24
5.3 BIM requirements Built2Spec use-cases.....	25
5.3.1 Thermal imaging	25
5.3.2 Air tightness	25
5.3.3 Acoustic and Indoor Air Quality	26

5.3.4	Thermal Imaging	26
5.3.5	3D model capturing (point cloud-BIM)	27
5.3.6	PHPP	28
6	Requirements to software/tools for quality checking on site	31
6.1	A BIM database with API.....	31
6.2	A BIM model checking language.....	31
6.2.1	State of the art	31
6.2.2	Conclusions	32
6.3	BIM IFC GUID integrity.....	33
6.3.1	Solutions.....	33
6.3.2	Conclusion.....	34
6.4	A WebGL viewer for BIM data	34
7	Integrate functionalities for BIM based quality checking.....	39
7.1	BIM Bots.....	39
7.2	Example Built2Spec.....	42
8	Automated quality checking of ad hoc design solutions versus predefined constraints	44
9	Issue management.....	45
9.1	Issue management and BIM	45
9.2	BIM Collaboration Format	45
9.2.1	Conventional BCF based issue exchange	46
9.2.2	New process.....	47
10	Conclusions	48
	References	49

Abbreviations

Term	Definition
API	API , an abbreviation of application program interface, is a set of routines, protocols, and tools for building software applications
BCF	The BIM Collaboration Format (BCF) is an open file XML format that supports workflow communication in BIM processes
BIM	BIM is an intelligent model-based process that provides insight to help plan, design, construct, and manage buildings and infrastructure.
BIMQL	BimQL (Building Information Model Query Language) is an open, domain specific query language for Building Information Models.
COBie	Construction Operations Building Information Exchange (COBie) is a data format for the publication of a subset of building model information focused on delivering building information not geometric modeling.
ESB	An enterprise service bus (ESB) is a software architecture for middleware that provides fundamental services for more complex architectures
GUI	The Graphical User Interface (GUI) is what a user sees on screen when using a software tool
IFC	The Industry Foundation Classes (IFC) data model is intended to describe building and construction industry data. It is a platform neutral, open file format specification that is not controlled by a single vendor or group of vendors.
JSON	JSON (JavaScript Object Notation) is a lightweight data-interchange format.
MVD	A Model View Definition (MVD) defines a subset of the IFC schema
Refurbify	Online project management tool (created and maintained by Built2Spec partner VRM) and the base of the VCMP.
SOAP	SOAP (Simple Object Access Protocol) is a protocol specification for exchanging structured information in the implementation of web services in computer networks
VCMP	Virtual Construction Management Platform.
XML	Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable.

1 Introduction

1.1 The Built2Spec project

Human-induced climate change is occurring globally and has already had a significant impact on the environment and society, due to increased levels of greenhouse gases. It is well documented that the built environment in Europe accounts for more than 40% of the overall energy consumption and 36% of the overall CO₂ emissions (Pérez-Lombard, et al. 2008). Furthermore, statistics show that construction and the upkeep of buildings and infrastructure is the largest industry worldwide accounting for approximately 10% of global gross domestic product (GDP) (ECTP 2012).

Concerning the problem of increased energy consumption in the building sector, the European Union (EU) adopted the Directive 2010/31/EU (European Commission, 2010). According to the Directive, member states are requested to adopt a methodology for calculating the energy performance of buildings (i.e. energy performance certification). The objective of this Directive (European Commission, 2010) is to ensure all new buildings are almost zero-energy consumption buildings by the end of 2020. In order to meet the requirements posed by the EU and, thus, reduce the environmental impact of buildings, energy efficient measures must be taken into account when designing new and retrofitting old buildings.

'Built2Spec - Tools for the 21st century construction site' project brings together a new and breakthrough set of technological advances for self-inspection and quality assurance that will be put into the hands of construction stakeholders to help meet Europe's energy efficiency targets, standards for constructing and retrofitting buildings, and related policy ambitions (source: built2spec-project.eu).

The centrepiece of the Built2Spec project is a cloud based virtual construction management platform (VCMP), conceived following the most advanced integrated design and delivery framework for the building sector. The VCMP will host applications that facilitate worksite activities and quality compliance by putting knowledge in hands of contractors. This will be done through shared design specifications and 3D models, installation guidelines, information on regulatory frameworks, and help from construction experts on smartphones and tablets, from initial design to delivery. The overall architecture for this VCMP is already described in D1.4 (System Architecture). D1.4 has been used as input for this D4.5.

1.2 Goal and context within the project

Within WP2 and WP3 partners develop advanced techniques for Thermal Imaging, Air Tightness, Indoor Air Quality and Acoustic testing. These tests may use input from a BIM dataset to perform measurements, and should be able to link results to objects in a BIM dataset. A generic workflow about this is shown in figure 4.1.

The goal of Built2Spec task 4.5 is "To make BIM technologically ready for self-inspection and quality checks".

The result of this task is a direct input for WP5 where the overall objective is to develop the detailed process steps for self-inspection and quality checks that a site worker and site supervisor would follow to acquire the data needed to automatically generate project files and reports which demonstrate compliance with the relevant standards, specifications and regulations.

1.3 Research topics

To make BIM technologically ready for self-inspection and quality checks within the context of the Built2Spec project, several research topics are discussed:

- 1) The (predefined) aim is to put all BIM data in a cloud BIM database. Within the project we will develop an integrated online (cloud ready) WebGL viewer using the BIM Surfer open source technology. This will allow to finally bring BIM to handheld devices for robust use on a construction site. This is being discussed in chapter 6.
- 2) To be used for self-inspection and quality checks, links to diverse/non-BIM content (defined in WP1, WP2 and WP3) to specific objects in a BIM model are being created. Technical and semantic interfaces will be created to optimize and automate the workflow of linking remote stored instructions (data) to BIM objects. This is discussed in chapters 3 and 6.
- 3) Being this a platform neutral solution, this task will also implement a methodology to avoid or handle BIM IFC GUID integrity problems over changing revisions. This is being discussed in chapter 6.3.
- 4) Integrate the functionalities for BIM based clash detection of different projects to be merged. This is part of the issue management, discussed in chapter 9.
- 5) Construction problems often arise on site, in which case the danger of rapidly designed solutions for encountered problems is that the solution may create new and potentially bigger problems in a later stage of the construction process. Short-term efficiency can generate longer-term ineffectiveness. Chapter 8 addresses this topic.

2 BIM

BIM is short for “Building Information Model” or “Building Information Modelling”. Sometimes the M is even explained as “Management”. In the often cited BIM Handbook (Eastman, et al., 2011) the first use of the term “Building Information Model” was attributed to two TNO researchers (Nederveen & Tolman, 1992). The principles behind BIM, as we know it today, are much older and are dated back to 1975. Despite the evolution of the name of the concept into BIM as we know it, the underpinning principles have not changed much.

Although the concept has been around for decades, there still seems to be a lot of confusion surrounding BIM in the building industry. In order to understand the principles, reading the aforementioned BIM Handbook, or even the Wikipedia article on BIM, is highly recommended. It is crucial to understand that BIM is about modelling a building in terms of data and that this is fundamentally different from drawing lines to represent a building, whether it is on paper, in 2D/ 3D using CAD software. With the uptake of BIM the building industry is changing from a paper-driven industry to a data-driven industry.

BIM adoption or BIM maturity, the extent to which stakeholders in the building industry use (the full potential of) BIM varies greatly between countries. In a recent study comparing the six continents, Europe was found to be the second in terms of BIM adoption, following North America (Jung & Lee, 2015). Within these continents there are still large differences between countries (NBS, 2016). Different countries have different strategies to promote BIM implementation (Smith, 2014). And within these countries the differences between companies can be very large. Some leading companies use BIM very effectively to work together on construction projects, whereas others do not use it at all. In addition, studies such as the BIM Quickscan in The Netherlands demonstrated that different stakeholders in the building industry show different BIM ‘levels’ (Berlo, et al., 2012). All in all, it is safe to say that BIM is not a fleeting trend, but BIM is here to stay, and its use and added value will only increase in the years to come.

2.1 Software Tools

Modelling objects into buildings (as opposed to drawing lines) requires software tools. In the 1990s Graphisoft started the ‘Virtual Building Concept’ with the release of ArchiCAD, and took the lead in the market for BIM software for architects. When Autodesk acquired Revit, the advanced distribution channel of Autodesk made Revit widely available in the industry. This instigated a huge growth of the generic concept of BIM in the architectural industry. In specialized disciplines of the building industry other modelling tools are used. A construction engineer will likely work with Tekla, Scia, or Allplan. Mechanical, electrical and plumbing (MEP modelling) is often done in MagiCAD or DDS. Tools like Solibri Model Checker are very popular to check the quality of the data and to coordinate different discipline models.

Most software tools express the same basic concept: a central database that is used to generate different views of the data, see Figure 2.1. When different stakeholders work closely together on a project, usually in less traditional contracts, they used to exchange files and compare them to each other. Currently a new wave of BIM tools is being developed where these exchanges take place online. Online BIM collaboration platforms are the new trend in BIM.

The number of BIM tools and start-ups keeps growing daily.

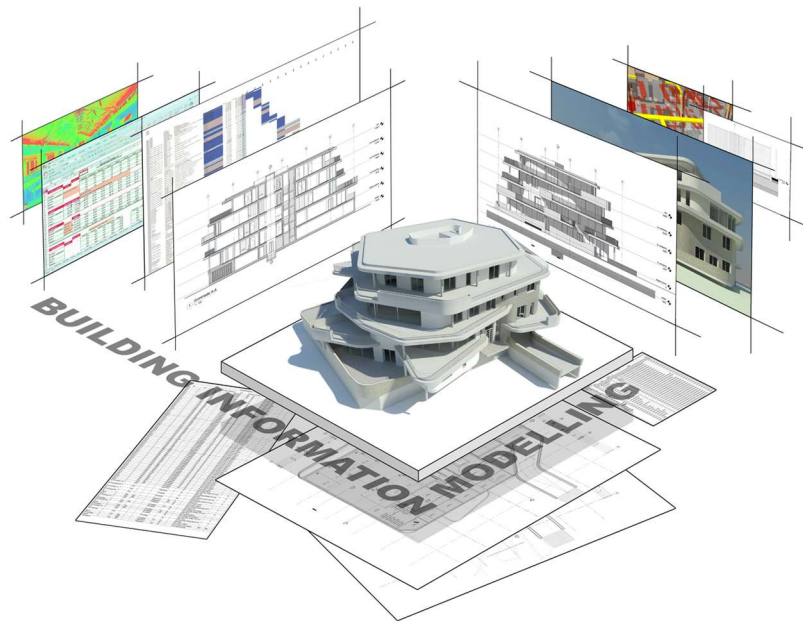


Figure 2.1 : A 'central' database to generate different views

2.2 Misconceptions

There are many misconceptions about BIM. Maybe the biggest misconception is that BIM is centralizing all data of a project into a single data repository.

Because most BIM software tools work with a central database that is being used for all features, this concept is being copied on a project scale. Recent rise of online BIM collaboration platforms is feeding this concept.

However, many research projects and publications have proven that working with a central data repository is actually decreasing productivity of the project (see Fig. 2.2 for a common dataflow in a BIM collaborative project). Working efficiently with distributed data storage is more effective for the project than trying to centralize everything.



Figure 2.2 : Common dataflow in a BIM collaborative project (Berlo et al, 2015)

In the slipstream of the urge to centralize data, there is a driving force to put as much data as possible in a BIM. When this is not possible, linking or connecting to objects in a BIM should be attempted.

There is a lot of research done about the concept of linking different data sources to data objects in a BIM. Not all research results agree with each other on the best approach. The field will change with the increasing use of online BIM platforms.

In figure 2.2 a common dataflow is given for collaboration with BIM. This result came from research by Berlo, Derks,, Pennavaire, & Bos about “Collaborative Engineering with IFC”. The authors concluded that the use of IFC in a downstream setup is more effective than proprietary data flows. The Built2Spec use-cases came to the same conclusion and within the Built2Spec project, the downstream IFC approach will be the main concept for data exchange.

2.3 BIM Data Standards

Every BIM software tool has its own internal data model. This is a data model that is most effective for the features of that tool. The data structure of every BIM tool is therefore different. When designers want to coordinate their designs with each other, a common data model to map the data from different tools is very helpful.

The most frequently used data standard for this purpose is IFC (Industry Foundation Classes). This data standard is developed and maintained by BuildingSMART International (<http://www.buildingsmart.org>). It contains around 800 objects and 12000 properties. All of which have a semantic documentation. This is very important for the interoperability of data in the industry.

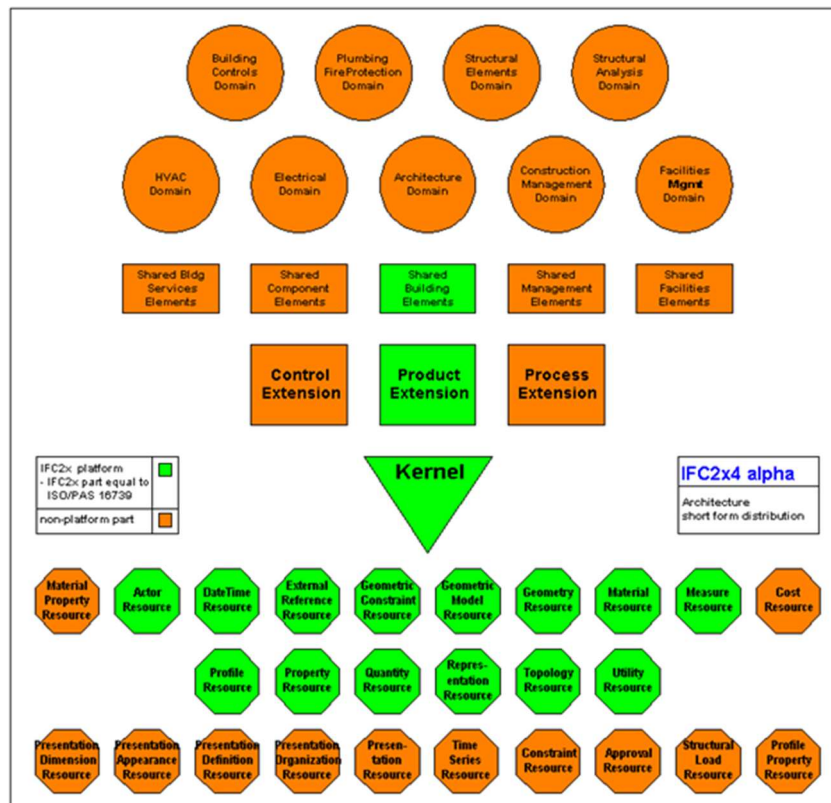


Figure 2.3 : Structure of the IFC data schema agreements

Most software tools that call themselves the ‘BIM tools’ have an IFC import/export function.

2.4 BIM Web Service Standards

BIM adaptation has not yet reached the necessary level for online interaction. In Built2Spec the online VCMP is a central element of the innovation. Therefore BIM also needs to move from the file-based data to online databases.

Many online BIM platforms are being developed at the moment. All of them have specific features and both positive and negative points.

The Built2Spec project used the open source BIMserver.org platform (<http://bimserver.org>) to get BIM objects in an online environment. The BIMserver initiative started in 2008 when the industry was still unaware of the concept of online BIM. The stability of the BIMserver.org platform is proven enterprise ready. Many commercial applications build on the stable base of BIMserver.

Since putting everything in one central data repository is not a solution by itself, the BIMserver (<http://bimserver.org>) has to interact with other online services. The API of BIMserver is based on the open API standard BIMSie (https://www.nibs.org/?page=bsa_bimsie) (Building Information Service Interface exchange). This open standard from BuildingSMART alliance makes it possible for online BIM platforms to automate interaction between each other.

2.5 BIM Issue Management

Within the Built2Spec issues management of BIM data is a very important part of the innovation. In an IFC dataset, every product has a unique identifier, but this ID is not meant to be human readable.

In practice, communication about objects in an IFC dataset is not user friendly and there is a large margin for human errors. This is why the ‘BIM Collaboration Format’ (BCF) (<http://www.buildingsmart-tech.org/specifications/bcf-releases>) is introduced.

BCF is a data standard for BIM ‘issues’. Technically BCF is a small standard. It holds not much more than a topic, description, a picture, IDs of the IFC objects with the issue, and coordinates of a camera viewpoint (how to best look at the issue). Each issue (which is called ‘Topic’ in BCF) can be assigned to a user, and classified with a priority, category, etc.

The use of BCF is not as widespread as IFC yet, but it is gaining a lot of movement and positive vibe. It is expected that the use of BCF will be just as high, or even higher than IFC in the near future.

BCF is very file based at the moment. All users e-mail zip files with BCF data to each other. The open source project ‘BCF Forum’ (<https://github.com/opensourceBIM/BCF-Forum>) (see Figure 2.4 for a screenshot) from the open source BIM collective tried to get BCF into an online environment. This experiment showed positive results that are reported in chapter 9.

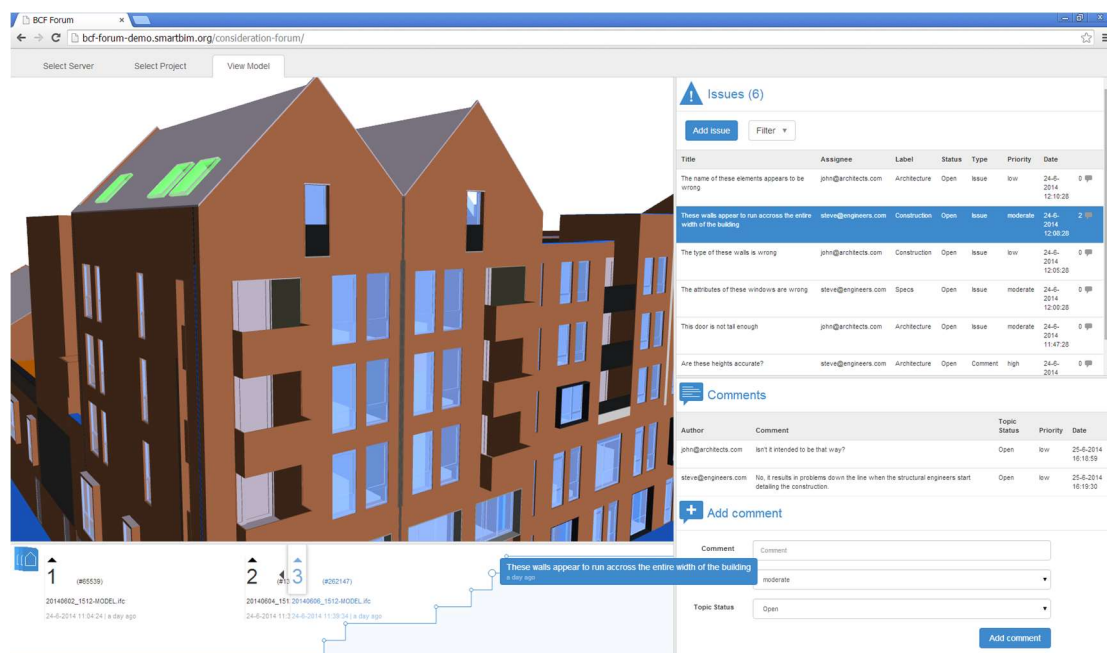


Figure 2.4 : The (online) BCF Forum in action

At this moment BuildingSMART is developing a technical interface (API) for BCF exchange between online BIM platforms.

Initial research showed that the BCF API is not stable and extensive enough yet for the intended use cases of Built2Spec. During the course of the Built2Spec project the BCF API will be monitored. A detailed assessment has to be made about the usability of the BCF API. This will be reported in D4.6.

3 BIM integration

3.1 Distributed approach

Connecting the online BIM tools to the other online parts of Built2spec is a great challenge. Recent developments in IT architecture tend towards decoupling of features. This so called ‘microservices’ movement is very promising. Microservices is a software architecture style in which complex applications are composed of small, independent processes communicating with each other using data-agnostic APIs. These services are small, highly decoupled and focus on doing a small task, facilitating a modular approach to system-building, see Figure 3.1.

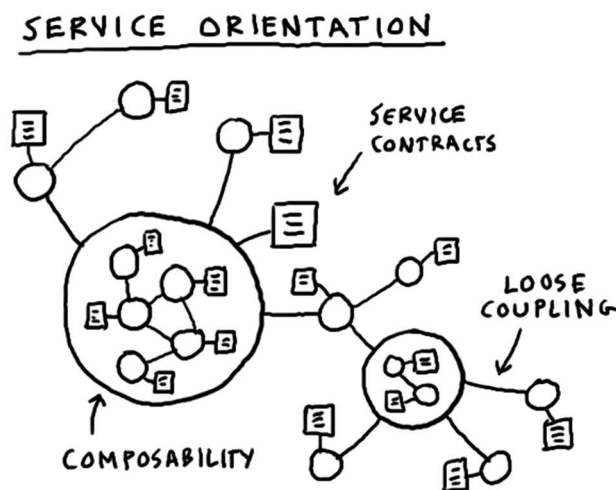


Figure 3.1 : The concept of Microservices

The chosen BIM applications (BIMserver, BCF Forum, etc.) are also focusing their development on the concept of microservices (see Figure 3.2). One implementation of this concept was introduced in (Berlo, 2014), was named “BIM Bots” in (Berlo, et al., 2015) and it has been a continuation of the BIMSie initiatives.



Figure 3.2 : The concept of BIM Bots (bimbots.org)

3.2 Integration of different systems

Within Built2Spec there are several different technologies, tools and systems that have to be connected. Recent developments in IT show promising results for the concept of ‘microservices’. The concept of “BIM Bots” harnesses the concept of microservices to the BIM and building industry.

A bot is a (online) system that triggers on an event. It performs a task with some form of “intelligence”. A BIM Bot can perform an analysis or simulation on your BIM data, it can enrich the model by adding detailed objects, splitting it for specific use, or perform any other task on your data.

The conclusion of D1.4 (Architecture of the VCMP) was to use the BIM Bots concept for the integration of the different technologies in Built2Spec.

This means that the VCMP will not be one monolithic system, but a combination of different microservices (‘bots’) that seamlessly integrate with each other. The core elements of the Built2Spec are the current VRM system, the BIMserver and the pointcloud service. The focus of the Built2Spec development should be on interaction between these main components of the VCMP. All Built2Spec technologies and simulation tools interact with one or more of these components.

4 Generic use case Built2Spec

4.1 VCMP generic process flow

The generic use-case in Built2Spec is described in the following picture (using an example of the air tightness test):



Figure 4.1 : The generic use-case in Built2Spec (using an example of the air tightness test)

- A user creates a task in the VCMP
- Additional information needed to perform the task is gathered and/or extracted from the BIM data
- A user might/could add (additional or required) information
- The user (or system) performs the task
- Results are stored and can be viewed

The storing of the results might trigger a new workflow. More information about this can be found in the deliverables of WP6.

4.2 Tools

Within the VCMP the following tools are being used:

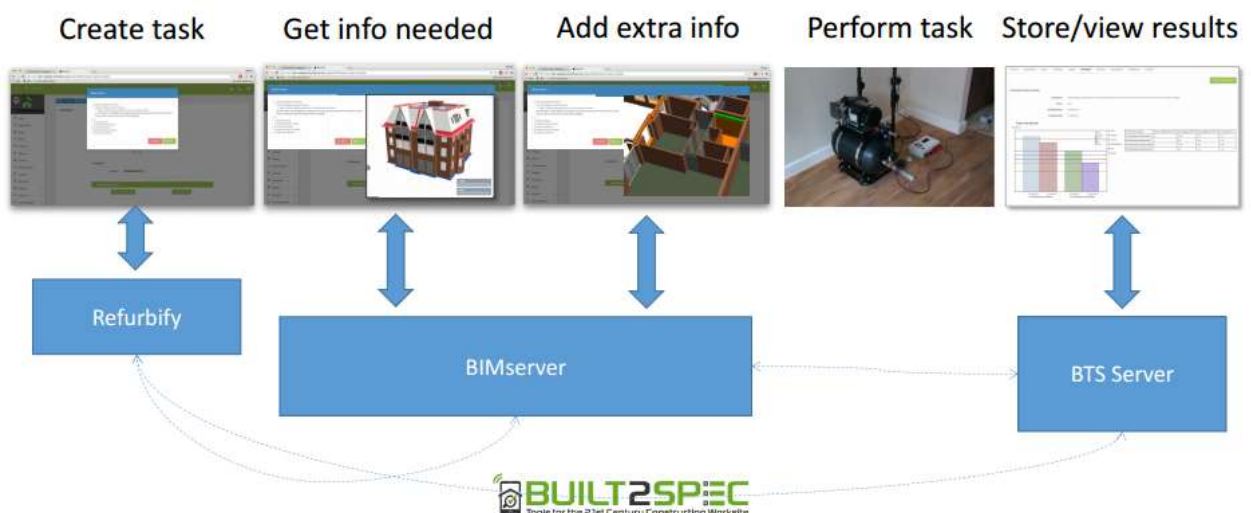


Figure 4.2 : Use of tools in the backend of the VCMP

- Refurbify for task management
- BIMserver for handling the BIM data in IFC
- Optional: A separate system/tool or server to store results

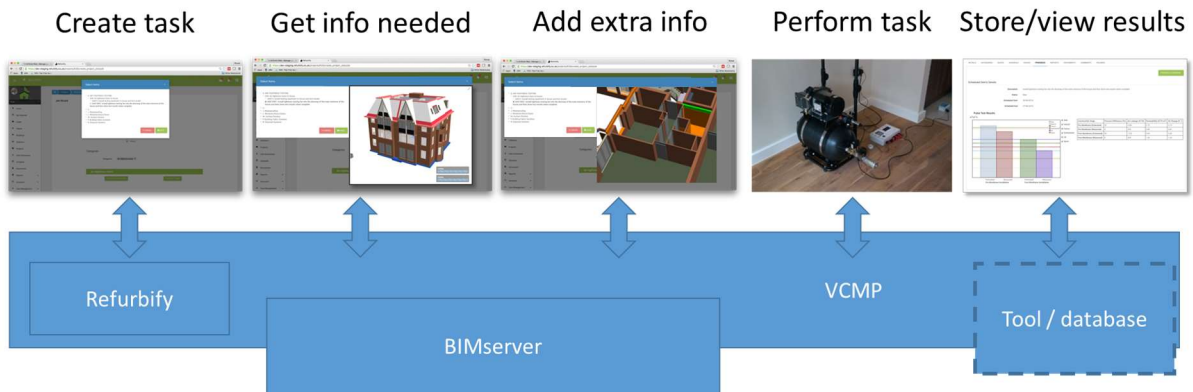


Figure 4.3 : Explaining the concept of the VCMP (connected system of several other tools)

The interaction with BIMserver is the most interesting part for this BIM report. When a new task is created, additional information needs to be linked to the task in order to allow the user to perform the task, or to automate the calculation of the results.

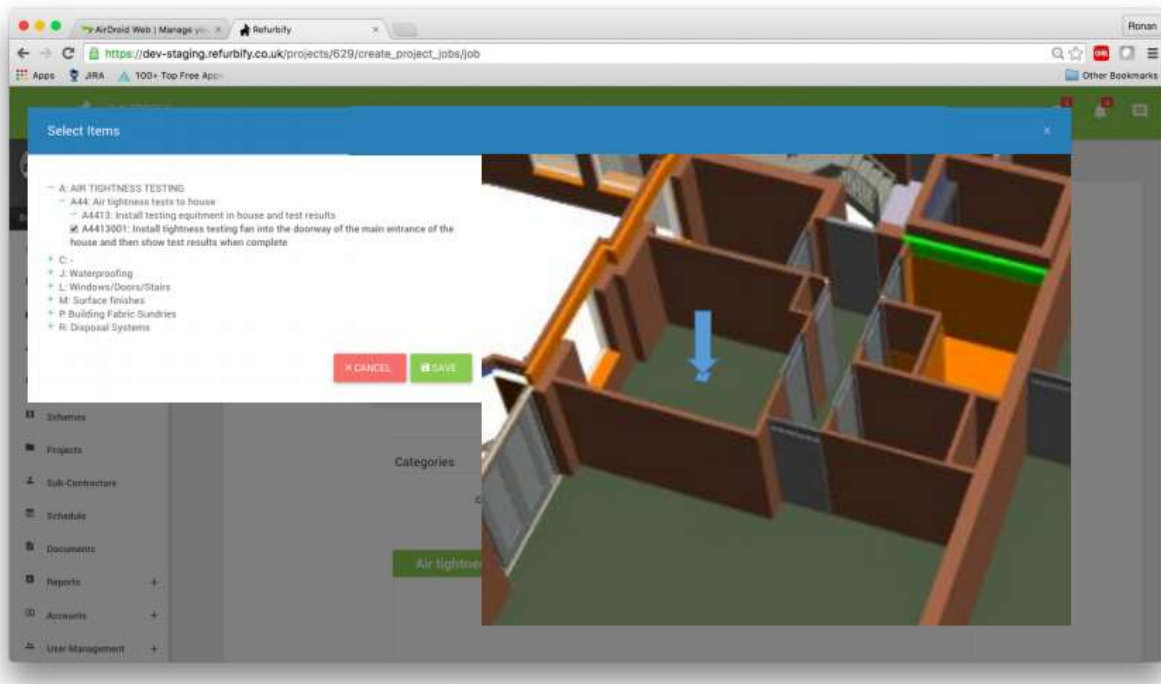


Figure 4.4 : Integrating BIM functions in the GUI of the VCMP

This could be 'location of the device' for the quality test; maximum deviation of the built elements compared to the design, outer area of the building, etc. Some of these requirements can be extracted from the model, some need to be added manually by an expert.

In chapter 5 we will detail what BIM data can be extracted to link to the tasks, and what the requirements to the BIM are to perform these tasks.

4.3 BIM technology and data flow

The underlying technology that is used during this workflow is a bit more complex.

The dataflow behind the generic workflow looks like this:

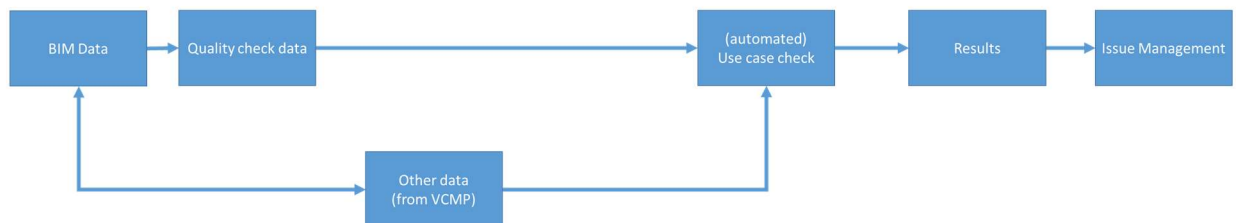
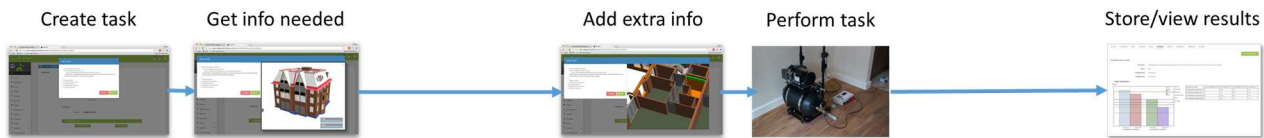


Figure 4.5 : What happens in the backend of the VCMP when operating it.

After creating a task, the information needed for that task is fetched from the BIM dataset (BIMserver database). Behind the scenes, the data are being checked on quality and completeness to perform the task.

Missing information is requested from the user. This information is stored in VCMP and is linked to the unique identifier of the BIM data (IFC GUID).

The task can be then performed (on site).

After the task is performed, the raw data of the measurement are being processed by the VCMP. In many of the self-inspection tests, the measurement data are processed and checked against the BIM data. The final results are presented to the user via the VCMP.

When issues arise from the results, they are processed in a next step referred to as ‘issue management’. Using BCF, these issues are being linked to BIM data.

To perform the above mentioned steps, additional technology is needed.

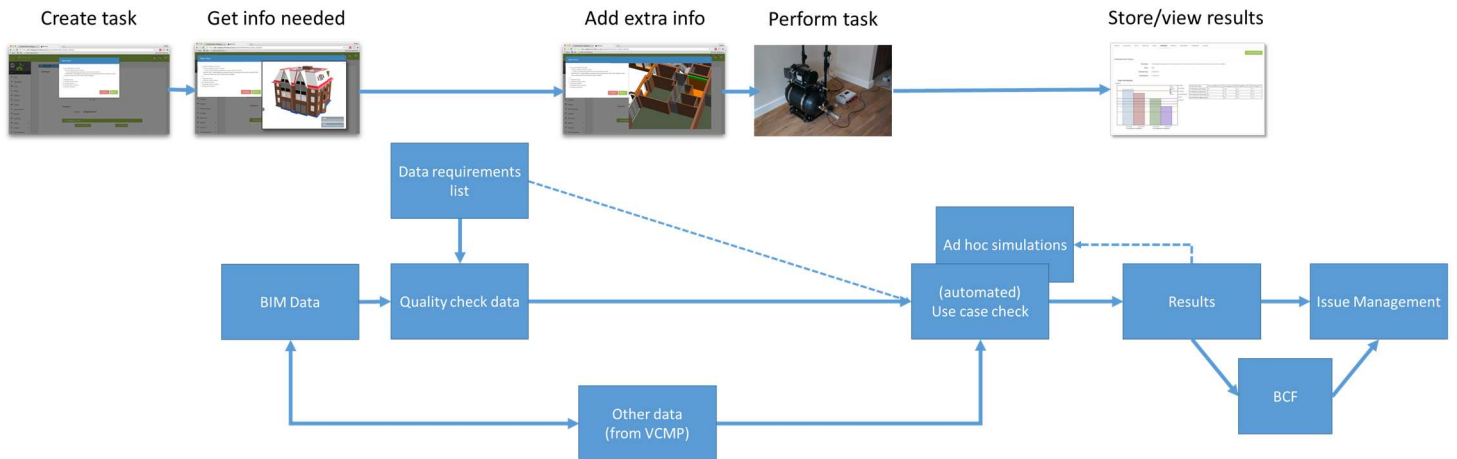


Figure 4.6 : Technology used to facilitate the operating of the VCMP.

The additional information included Data Requirements for BIM, additional simulations and BCF styled issues. The report will detail the technologies in the next chapters.

4.4 Example

In this chapter an example use-case is explained in more detail. This example is the point cloud use-case lead by ETH.

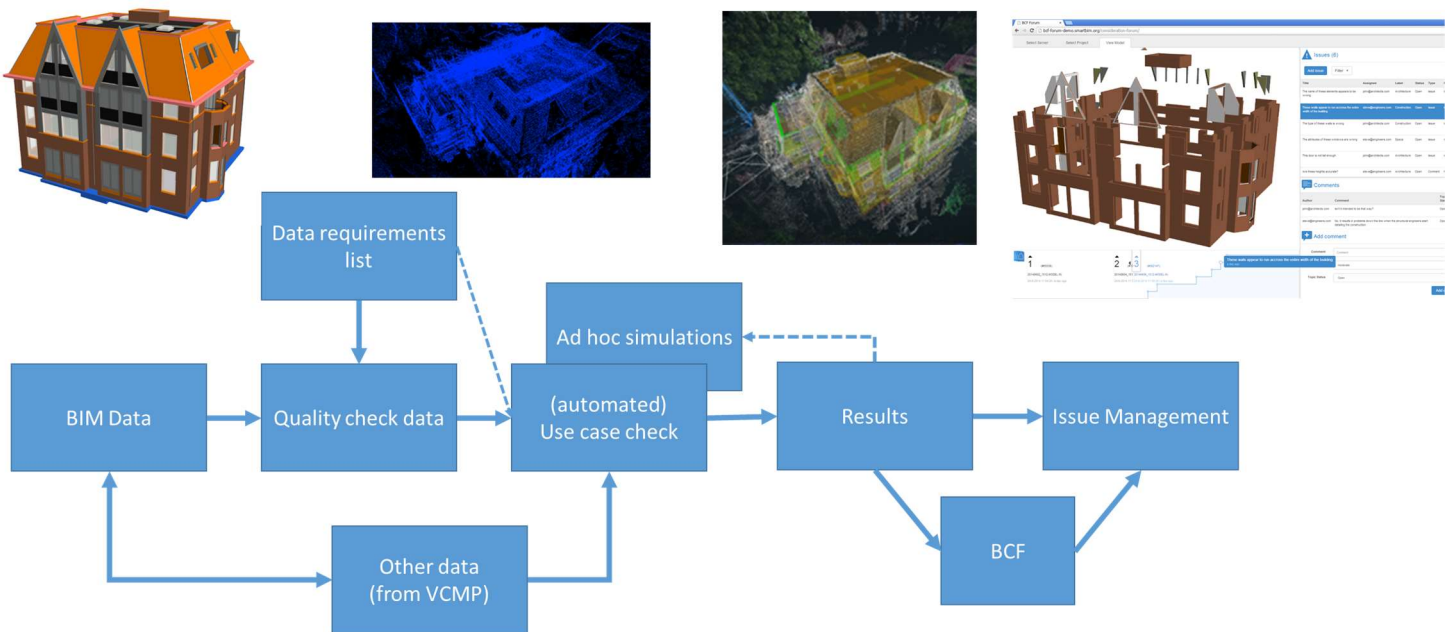


Figure 4.7 : Example of the technologies and dataflow used to facilitate the Point cloud self-test use-case of Task 4.3

In this use-case, a user walks on the site whilst using 3D scanning techniques (more details in D4.3), to create a point cloud of the actual building site. The point cloud is then compared with the original

design in BIM (more details will be provided in D4.4). Differences between the actual constructed object, and the designed BIM object are being highlighted and processed in VCMP.

A step by step process of the technologies used:

- A BIM model in IFC is queried from the database (BIMserver)
- The IFC data are checked against a predefined requirements list. The requirements per use-case, and the technology to store these requirements are being detailed in Chapter 5 of this report. In this case the requirements to IFC are for example 'objects have to have a 3D geometry', 'the location of the elements needs to be exact', 'the type of elements needs to be added (walls, windows, slabs, etc)', 'materials should be added to the objects using classification list x', etc.
- Other data, linked to the IFC data comes from the VCMP database. A list of construction margins per object is being queried. This list states the maximum allowed deviations per object type. For example columns may deviate +/- 2 mm from the original design; walls may deviate +/- 10 mm; etc. In some cases, this additional information is being supplied by the expert user during the creation of the task.
- Using these data, an automated check is being performed. In this case the BIM geometry is aligned with the point cloud (using the ETH algorithm which is detailed in D4.3); objects are being recognized; deviations are being calculated; etc. This calculation is usually done in the cloud because hardware on site is not capable of performing these calculations. An internet connection is needed to start these simulations on site.
- Finally, a result is being generated and shown to the user. Per case the result is different and it can be presented as a number, a graph, a picture, a 3D model, etc.
- Depending on the result of the checks, an 'issue' is created. These issues are being processed using BCF. The BCF issue can be stored in BIMserver or in VCMP and is linked to the original IFC object.

4.5 Reading guide for next chapters

The next chapters will detail the Built2Spec BIM approach:

Chapter 5 lists the requirements to BIM data both general and per use-case of Built2Spec.

Chapter 6 describes the available technologies that can be used to semantically store these requirements so they can be checked by a computer. This chapter highlights both data standards and tools that can be used.

Chapter 7 highlights technology that is used to automate the checking.

Chapter 8 is a side step in the process, and describes the extendibility of automated checking for ad-hoc checks on site, and early process simulations.

Chapter 9 describes how issue management is handled using BCF and BIM in an online environment.

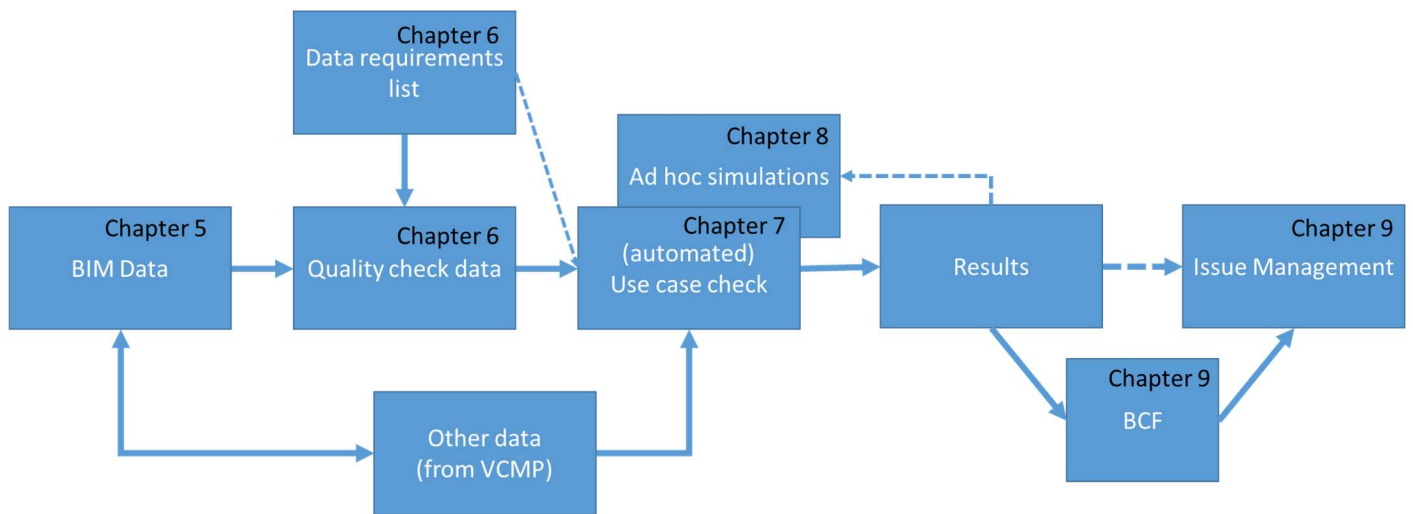


Figure 4.8 : Reading guide for the technology chapters of this report

5 Requirements to BIM data for quality checking on-site

This chapter will describe the general BIM requirements for quality checking with BIM.

In order to guarantee an interoperable use of IFC models, requirement documents are being written that make statements on the validity or appropriateness of (parts of) building model definitions. These use cases range from checking the conformance to rules set out by (local) governments to in-house guidelines and Level Of Development (LOD) specifications. In a similar fashion, the compliance to building codes needs to be checked in order to obtain permits. Several terms have been coined to name such requirements, such as “BIM norms”, “BIM employers’ requirements” and “BIM protocols”. They govern the requirements that people, organizations and government pose on the data or edifice that is delivered to them. Being able to check these requirements in an automated way is highly desirable for effective data exchange and high quality end-results (Berlo & Krijnen, 2016).

5.1 BIM Standards

BIM equals data. It is a collection of virtual objects with properties and relations. Because a computer has semantic awareness of the objects, intelligent operations can be performed on the data. The richer and more semantic the dataset, the more intelligent the operations can be.

Every BIM software tool has its own internal data model. This is a data model that is most effective for the features of that specific tool. The data structure of every BIM tool is therefore different. When designers want to coordinate their designs with each other there needs to be a common data model to map the data from the different tools.

The most used data standard for this purpose is IFC (Industry Foundation Classes). This data standard is developed and maintained by BuildingSMART International. It contains around 800 objects and 12.000 properties. All of which have a semantic documentation. This is very important for the interoperability of data in the industry.

Every software tool that calls itself a ‘BIM tool’ has an IFC import/export function.

In the IFC ecosystem other data standards are also interesting:

- mvdXML to define ‘Model View Definitions’ that filter a part of IFC for specific applications;
- ifcXML (same as IFC, but different syntax)
- simple ifcXML (same as ifcXML, but less overhead in the syntax)
- COBie (well known MVD specifically for Facility Management)
- BIM Collaboration Format (BCF) for issue management

Other non-Buildingsmart BIM data standards are BIMxml, gbXML and STL. All of these have little traction compared to IFC.

Different use cases (might) need different structured IFC input data. For example the thermal imaging check needs to know what kind of objects are in the picture. For this check there needs to be a semantic difference between different objects like a wall, a window, isolation, etc. The air

tightness test doesn't need most of this, but needs to have information about the area and volume of a space.

There are several ways to further detail the requirements but the two most common are:

- Classifications
- Model View Definitions

5.1.1 Classifications

In IFC a wall is modelled as an 'IfcWall' object. This can be any kind of wall. To detail the semantics of a space it can be classified as an 'inner wall' or 'outer wall', or even more specific 'out wall between two buildings'. This classifying can be done with terms that the BIM modeler just makes up, or a project team agrees to use one specific. For efficiency reasons standardized classification references are used. Commons classifications are the 'OmniClass' (US), 'Uniclass' (UK) and in the Netherlands 'NISfb'.

Within the Built2Spec project, we decided to use the OmniClass classification schema, since it is loosely integrated in the most commonly used BIM Tool Revit.

5.1.2 Model View Definitions

The IFC data schema is very extensive and rich. It has about 800 objects and 12.000 properties defined.

The full extent of the IFC agreements are never implemented in software tools. Software vendors only support a part of the full IFC agreement. To define what parts the concept of '*Model View Definitions*' (MVDs) is created. An MVD defines which parts of the full IFC agreement are supported. The most used and infamous MVD is the 'coordination view'. Almost every IFC export from a major BIM software tool exports IFC according to the Coordination View MVD.

The use-cases in Built2spec also have specific requirements. We will go into that in paragraph 5.3 but to give an example: the air tightness test needs to know the volume and area of the building in a specific way; and the comparison between the point cloud and the original BIM can only handle tessellated geometry (without Boolean operations). This is a kind of MVD, although it is not officially registered as a BuildingSMART MVD (note: BuildingSMART is the owner and developer of IFC and the official MVDs).

To define a specific MVD a new standard is in the making: mvdXML. This is an XML syntax to define a specific MVD for your own software tool. Datasets can then be checked against the mvdXML file to validate if all the data is available for the software tool to perform.

At the moment we found mvdXML to be too limited for the use-cases in Built2spec, so we decided to use other tools to check the BIM requirements. More on that in Chapter 6.

5.2 General BIM Data requirements

To use the information in the use-cases, it needs to be in IFC, and therefore someone needs to model it in the BIM modelling tool.

The general requirements to the IFC datasets are:

- IFC version 2x3
- Split up the elements in as much objects as possible
 - o so as little 'assembly' as possible
- use the correct elements of IFC for the objects (so no proxys)
 - o for example: use 'IfcWall' for walls, 'IfcWindow' for windows, etc.
- IfcTypeObject should be used for elements
 - o and put generic properties in the type object
- IfcMaterial has to be added to the elements
 - o layered materials are allowed
- The data/model should be free of internal clashes or duplicate objects
- A reference point on the actual site should be modelled
 - o add a 'built2spec reference' classification to it.

If possible, the following requirements will improve the change of high quality results:

- Second order space boundaries
- Use of loadbearing Boolean properties
- Use of IsExternal Boolean properties
- Use of a classification reference to each object
- Use of the 'common' property sets to elements

Most of these requirements are compliant with the broadly used "Coordination view 2.0" Model view (MVD).

5.3 BIM requirements Built2Spec use-cases

The generic BIM requirements handle most of the structure of the data. In the Built2Spec use-cases there are also requirements to the content of the BIM data.

5.3.1 Thermal imaging

This use-case requires the BIM data to be modelled with semantic types. For example modelling walls as 'IfcWall', slabs as 'IfcSlab', etc. These requirements are already covered by the generic BIM requirements.

5.3.2 Air tightness

The air tightness test needs the following information:

- Volume of the 'conditioned part' of the building
- Area of the surrounding envelope of that conditioned part
- Building identifier like address or geo coordinate
- The stage of the construction
- The predefined air tightness level

The first three could be obtained from a BIM dataset. The last two (stage of construction and predefined air tightness level) should come from Refurbify/VCMP or an expert user.

The requirements are translated to BIM requirements:

<p>volume of the 'conditioned part' of the building</p>	<p>The 'conditioned part' needs to be identifiable in the IFC dataset. This can be done using classifications or using 'zones'. In Built2Spec we propose to use 'IfcZone'.</p> <p>An IfcZone is an aggregation of spaces, partial spaces or other zones. IfcSpace's are aggregated into an IfcZone by using the objectified relationship IfcRelAssignsToGroup as specified at the supertype IfcGroup.</p> <p>By using 'IfcZone', the Pset_ZoneCommon is a standardized set of properties that can be linked to the defined zones.</p> <p>Two properties in this set are 'GrossAreaPlanned' and 'NetAreaPlanned' that describe the total planned net and gross area for the zone.</p>
<p>area of the surrounding envelope of that conditioned part</p>	<p>The area of the surrounding envelope will be calculated by an algorithm in one of the BIM tools. This data is not explicitly stored in the IFC data, but derived using a software tool.</p>
<p>building identifier, e.g. address or geo coordinate</p>	<p>It is an option to use the IfcAddress (connected to the building) object type or SiteAddress property (connected to the site) for this. However, since addresses are not standardized most of the time, we propose to use RefLatitude, RefLongitude, and RefElevation to store the geocoordinates of the site. Lat and lon are defined as integer values for degrees, minutes, seconds, and, optionally, millionths of seconds with respect to the world geodetic system WGS84.</p>

5.3.3 Acoustic and Indoor Air Quality

For the acoustic and indoor air quality tests, the following information can be retrieved from the BIM Dataset:

- Surface of the wall between two rooms (separation wall)
- Volume of both rooms of measurements

For this information to be exact, the IFC should be exported with the option of 'second order space boundaries'.

5.3.4 Thermal Imaging

For the thermal imaging quality check, the following information can be retrieved from the BIM dataset:

- Thickness of each layer in the component (component meaning wall, window, etc.)

- Material per layer (preferably including the U-value, otherwise stored using a predefined classification system so the U-value can be extracted from an open database).
- Preferably transform the geometry to import the model in a platform such as Vuforia.

For these requirements, the multiple layers per object have to be exported in IFC. This is a different setting per BIM software tool so it could be checked when the data comes in the database.

5.3.5 3D model capturing (point cloud-BIM)

The 3D capturing use-case is in its core comparing the situation on site (as a point cloud dataset) to the original design in IFC.

This use-case is actually an aggregation of several different use-cases:

- Comparison between point cloud and IFC during fabrication of elements in the factory;
- Comparison between point cloud and IFC on site
- Comparison between point cloud and IFC after installation (on site)

For all of these sub-cases the following information is needed:

- The type of elements (walls, windows, slabs, etc.)
- Geometry of the elements
- Location of elements
- A reference to a fixed point
- Optionally: the material of objects (it might be the case that the creation of an accurate point cloud model from images need to know from what kind of material the object is made)

The requirements are translated to BIM requirements:

The type of elements (walls, windows, slabs, etc.)	Use the correct elements of IFC for the elements. For example: use 'IfcWall' for walls, 'IfcWindow' for windows, etc. Additional requirements could be made for a classification system. No proxy elements should be in the data.
Geometry of the elements	The geometry of the elements should be modelled in detail in the BIM system, and exported in the IFC dataset. The VCMP (using BIMserver.org) will perform Boolean operations on the geometry and store the resulting triangles (including tessellation) in the database.
Location of elements	The users needs to position the elements in the model relative to the reference point.

	The location of elements in IFC is usually stored relative to other elements. The BIMserver.org tool can extract the location from the origin or reference point in the model.
A reference to a fixed point	To make a valid comparison, the point cloud and the IFC data need to have a shared reference point in both datasets. This reference point could be any object with geometry and is different per pilot site. The object needs to have 'Built2Spec reference' as a classification in the IFC data.
<i>Optional:</i> material of objects	IfcMaterial has to be added to the elements. Additional requirements could be made for a classification system.
<i>Optional:</i> Schedule of the construction	To know which objects should be on site at a given time, and which shouldn't.

5.3.6 PHPP

One of the main tasks by PHI is the export from BIM project data to the PHPP in WP6. The evaluation of this interface provided two possibilities:

Interoperability using plug-ins

The Application Programming Interface (API) of a BIM authoring tool allows independent software developers to create programs that can access elements in the 3D model. The API is therefore something like a very strict "convention" or "blueprint" of how the plug-in needs to communicate with its "parent" application.

The main drawback is that Plug-ins have to be programmed individually for each BIM software application because they offer different APIs. Due to the rapid development of the BIM software industry, newer versions of programs can have interoperability issues with its plug-in. Therefore the IFC data standard for BIM is used.

In order to achieve interoperability with IFC and improve the efficiency of data transfer between BIM software applications and the PHPP, the Passive House Institute must develop an IFC compliant application tool that allows the conversion of data from an IFC format to Microsoft Excel.



Due to the fact that as many BIM programs as possible are to be supported, the best strategy seemed to be is to use the IFC format file in version IFC4 Reference View for data interoperability with the PHPP. Indeed, compared to other file formats this has prevailed as an open standard and is supported by a large number of BIM authoring tools such as Revit, ArchiCAD, Vectorworks or Allplan

and more. In addition, the IFC format file can also be extended independently with custom parameters, which suits the specific PHPP requirement.

In this setup there are two main challenges:

1. The BIM software does not provide the level of detail required for PHPP

In many cases, the BIM modeler does not input foreseen parameters which are needed for a detailed energy balance calculation, as for example the Passive House Planning Package (PHPP). For example for windows, the width and height of the window are available by default, but a more detailed U-value or the thermal bridge of the window installation is not entered into BIM software because there is no need for the modeler to perform that task. Same goes for any additional information on thermal bridges, although it is highly relevant for high efficiency projects to consider this thermal bridge. So PHI had to find a way how to include these parameters into the native BIM dataset (Revit / ArchiCAD)

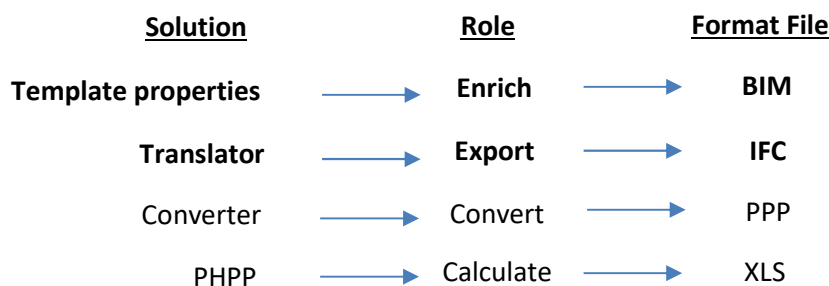
2. The BIM software doesn't export parameters to IFC

Some parameters required for an energy balance calculation are available in native BIM datasets or respective energy efficiency modules, but they aren't exported to the IFC dataset. For example the lambda value of building assembly layers would be editable in the BIM authoring tool, but not exported to IFC by default.

Solution: USE BIM templates to make available efficiency parameters

After evaluating the advantages and disadvantages of plug-ins, PHI decided to implement missing efficiency parameters by setting up component templates and make these available for the main BIM Authoring tools (REVIT / ArchiCAD). So these templates for walls, roofs, floor surfaces or windows where set up, already including the parameters which are required for the energy balance calculation.

These parameters are then exported into IFC as user defined parameters, which can be identified by a converter tool and then exported to PHPP. This converter from IFC to PHPP is developed in WP6, the whole process of implementing energy efficiency data into BIM software and then make it available in PHPP looks as follows:



The overall dataflow of the PHPP, VCMP and BIM is defined in Figure 5.1:

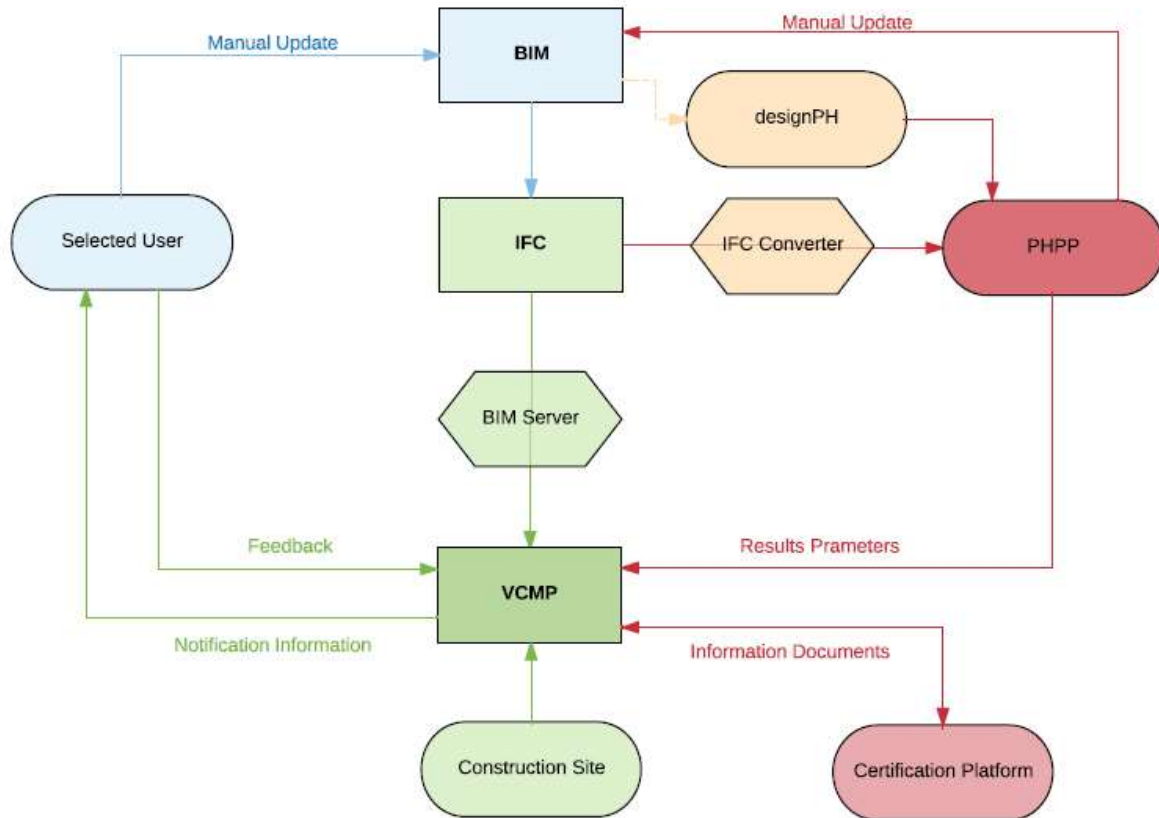


Figure 5.1 : Dataflow between VCMP, BIM and PHPP

6 Requirements to software/tools for quality checking on site

This chapter describes the requirements of BIM software and tools for self-inspection and quality checks in general and the use-cases specific to Built2Spec.

6.1 A BIM database with API

BIM adaptation is not on the level yet where online interaction is considered to be a necessity. The next step in BIM is the shift from file based data to online databases.

Many online BIM platforms are being developed at the moment. All of them have specific features and both positive and negative points.

In Built2Spec we decided to use the open source BIMserver.org software.

BIMserver.org is an open and stable software core to easily build reliable BIM software tools. It was created as a foundation for developers' applications. BIMserver has core server features like revisions, authorization, compare, query, model checking, merging, etc. The threshold for developers is as low as possible: BIMserver has lots of open interfaces and network protocols (soap, json), uses open standards, is built as a plugin framework for easy fine-tuning, has a flexible admin configuration GUI and very good developers documentation and software development kits (SDKs).

The core of the software is based on the open standard IFC and therefore knows how to handle IFC data (both IFC 2x3 as IFC4). The BIMserver software is not a fileserver; data are interpreted and stored as objects in an underlying database. The main advantage of this approach is the possibility to query, merge and filter the BIM data. There are many possibilities to connect to the software (interfaces, API's). This makes it possible to write your own (closed source) application but still use the BIMserver platform underneath. The core of the BIMserver software is published with the GNU Affero GPLv3 license. (BIMserver.org, 2017)

6.2 A BIM model checking language

On a technical level, various approaches are being researched to assess BIM requirements on actual models. These approaches allow for different levels of expressivity and extents to which they allow for automation, modularity and reuse. This chapter will give an overview of available technical solutions to automate data requirements checking. It will state characteristics, advantages and limitations pertaining to these technologies. The technologies included in this overview are the IFC schema and its implementers agreements, Model View Definitions (mvdXML), classification systems and concept libraries, query languages, reasoners and proprietary software solutions.

The findings of this chapter are also published by Berlo & Krijnen (2016).

6.2.1 State of the art

Various model checking platforms exist and are described in literature or are commercially available. A platform used in practice is Solibri Model Checker (SMC). It is a JAVA-based executable that reads an IFC model and provides proprietary processing routines to facilitate the rule checking on common operations in the Architecture Engineering and Construction (AEC) industry. This includes checking for the existence of attributes (so called pre-checking), and more advanced fire exit and evacuation

checking and comparisons against the schedules and design briefs. SMC is a proprietary application that implements rules by means of hard-coding them in program code, therefore, other than adjusting specifically targeted parameters, this program is not extensible to add new types of rules.

Jotne EDMModelChecker (EDM) is a commercial library that separates the definition of rules from the program code. Rules are defined in the EXPRESS modelling language, an ISO certified open standard, in which the IFC schema is conceived as well (Eastman, et al., 2009).

FORNAX is among the first large government-involved effort towards automated rule checking in the building industry. It is part of the Singapore CORENET platform, developed as an automated system to regulate building permits. It is implemented on top of the EDM Model Checker (Khemiani, 2005).

An attempt to aid the formalization process of rule definitions is provided by SMARTcodes, which presents methods of converting codes and standards from textual natural language definitions into computer code. This is accomplished by means of semantically structured domain knowledge (Nawari, 2012).

Automated approaches to validate the conformance of a model to a Model View Definition (MVD) are described in (Zhang, et al., 2013). But as described in (Solihin, et al., 2015), what constitutes a valid, meaningful and unambiguous exchange is broader than what currently can be expressed in such a MVD and includes in addition aspects such as geometrical and topological correctness, for example that spaces are correctly bounded and that the faces that constitute this boundary conform to their (typically planar) underlying surface geometry.

Disambiguation is a crucial part of the formalization of rule definitions. For this purpose the use of multilingual concept libraries is crucial as it allows to unambiguously point to a well understood concept from within diverse national classification systems and different languages (Palos et al., 2014). In the AEC industry initiatives are being undertaken to implement such concept libraries specific to the industry, such as the buildingSMART Data Dictionaries (bsDD). In addition, well established ontologies with a broader scope are available, for example (Navigli and Ponzetto, 2012) and (Miller, 1995).

An orthogonal line of research is initiated by (Krijnen and Tamke, 2015) that tries to employ machine learning concepts, such as anomaly detection to enable model checking without the à priori definition of formal rules, but instead deduce a norm to which most building elements conform to and flag the elements that deviate from this norm.

Eastman, et al. (2009) describes a variety of rules and discusses the implementation of them in various of the available rule checking platforms. Often these rule sets are a mixture of safety or programmatic requirements. This work identifies major issues, such as a lack of extensibility of some of the platforms and resonates the finding that none of the platforms address the entire scope of rule checking. This entails the process of converting rules from natural languages into formal definitions, pre-checking the suitability of the model for more rigorous checking, executing the checks that often need geometrical and topological functionality to abstract building models into spatial structures suitable for e.g. fire exit checking and finally reporting the results.

6.2.2 Conclusions

In order to come to an efficient automated rule checking workflow, the technical aspects discussed by Berlo and Krijnen (2016) need to be viewed holistically and not in isolation. By exploiting their complementary nature, rules can be encoded formally, succinctly and expressively.

A crucial foundation for rule checking are the semantics introduced on the schema level: in order to prevent false positives and negatives, elements need to be classified correctly. Currently there are no or very limited provisions in the IFC schema that actually guarantee this formally. As a consequence elements are misclassified, or classified in overly broad or meaningless categories, such as `IfcBuildingElementProxy`. Elements that are misclassified result either in false negatives, as certain relevant checks are not performed, or in false positives, because irrelevant checks are performed. Both are detrimental to the quality of the rule checking process and therefore to the delivered artefact. For this purpose, a large body of the documentation of the IFC standard, which consists of hundreds of pages, needs to be translated into formal and decidable statements.

Secondly, formal notations of explicitly encoded quantities are necessary in order to guarantee that the available quantities that are extracted from the dataset are accurate. IFC is a highly redundant data format: the length of a wall will typically be reflected in many places within the schema, including space boundary geometries, explicitly calculated quantities and different representations for the body and axis of elements. The quality of a data schema can be described by means of a minimal redundancy and maximum reliability (McLeod, 1995). Without formalized connections between the different apprehensions of attributes, like the wall length, the amount of redundancy induces an increased risk of errors when parts of a model get updated (Berlo & Krijnen, 2016).

6.3 BIM IFC GUID integrity

In the BIM research field, the problem of 'GUID integrity' is well known. Every time a BIM Authoring tool exports IFC, the Global Unique Identifiers (GUIDs) might not be the same anymore for the same objects. This is due to the fact that many exporters of BIM Tools generate the GUIDs on the fly during export.

Due to this, every new revision of an IFC dataset has 'all new' objects when a computer analyses the objects based on the GUIDs. IFC documentation states clearly that objects that don't change should keep the same GUID, but in practice this is clearly not the case.

When working with a 'Common Data Environment' (CDE), this creates a problem for advanced model analyses (like merging, model fusion, checking, etc.).

6.3.1 Solutions

The theoretical solution for this issue is to improve the quality of the export of IFC from the BIM Authoring tools (and in the slipstream, also the import functions).

Theoretically, according to the IFC documentation, the use of 'IfcOwnerHistory' should also be used to manage the changes in an IFC dataset.

Although import/export is improving, perfect implementations are unrealistic in practice. Use of 'IfcOwnerHistory' is never seen in IFC data from practice by the authors of this report.

We conclude that a theoretical solution is not realistic at the moment.

Fixing the issues, by analyzing an IFC dataset is also not possible. This is due to the fact that IFC has a network structure, uses implicit 'inverse relations' and has many different ways to describe the same object (authoring tools choose themselves; mostly without user input).

On the practical side, users have found several ways to avoid or ignore the issue.

- Good collaboration workflow could mean that the lack of overlap in GUIDs is not a problem at all (reference model concept)
- Geometry matcher (shape and location) -> same resulting geometry can be described different semantics; matching should be on tessellated geometry
- Matching semantics (same semantic meaning and properties) -> doesn't work on new revision with changes, and doesn't work between aspect models

Other potential practical solutions that haven't been proven in practice yet could be:

- Graph matching (change detection)
- Use of BIM Collaboration Format (using IFC Snippets)
- New standard for exchanging only changes (changeset / API)

6.3.2 Conclusion

In Built2Spec, the project team decided to ignore the GUIDs and communicate based on ObjectTypes, Classifications and Assembly Codes.

Using the Assembly Code property of an IFC object gives the users flexibility to store a user-defined identifier, replacing the automatically generated GUID as the main identifier of an object. Standardizing a software tool on an Assembly Code shifts the workload to the users instead of the software, but since users can automate this in many of the BIM Authoring tools, this solution seems to be acceptable.

A more extensive report of the GUID integrity, and the possible and proven solutions based on Built2Spec, will be published in a journal paper.

6.4 A WebGL viewer for BIM data

To have the BIM data available on a construction site, it is recommended to have a viewer that can be used on mobile devices. Since there are many different mobile devices, and the graphical performance of most devices is not very high, this is a challenge.

A common way to deal with this issue, is to use cloud based performance and stream the results to the mobile screen. In this way, the mobile device is only viewing the content, while the advanced calculations are performed by large cloud computers.

Within Built2Spec the open source BIM Surfer tool is used as an online viewer. This software is based on WebGL technology. It can be found on www.bimsurfer.org. The source code is available on <https://github.com/open-source-BIM/BIMsurfer/>

The BIM Surfer initiative is part of the open source BIM collective, which also includes BIMserver.org. This means BIM Surfer and BIMserver are working in close collaboration to deliver an integrated solution.

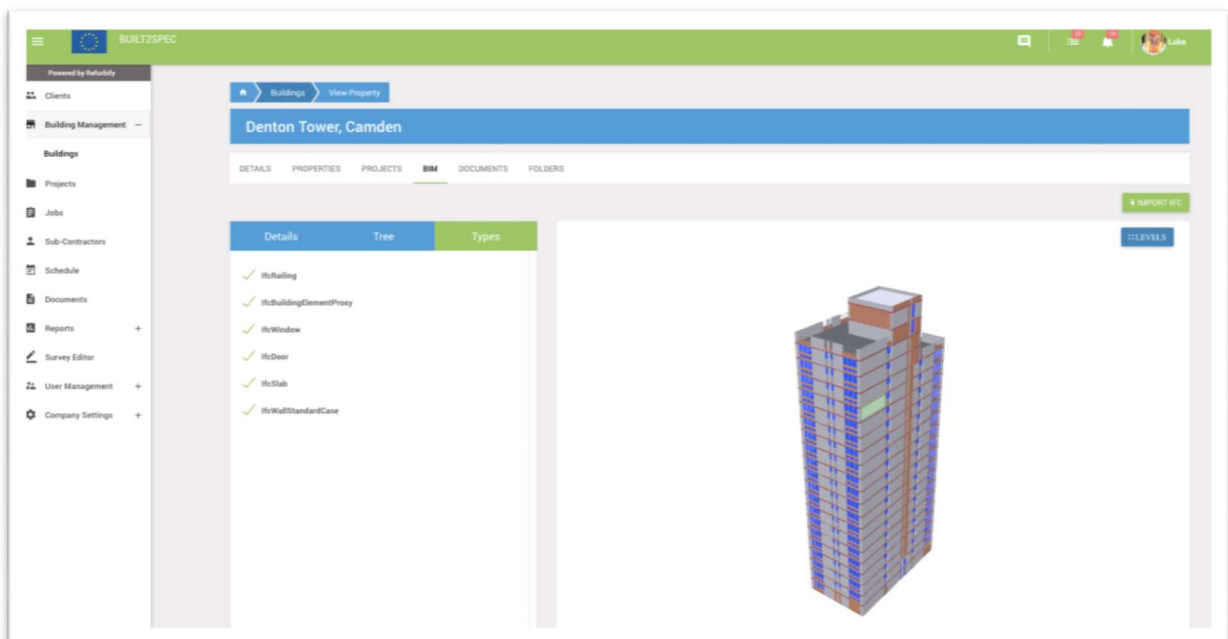
The focus of the BIM Surfer initiative has shifted from being a feature rich viewer, to a software development kit that can be integrated in several systems to enrich those tools with BIM viewing capabilities. This process was mainly guided by the Built2Spec project. As a consequence of this shift in focus, the license of the BIM Surfer also changed to MIT, making it more flexible to be used in proprietary systems.

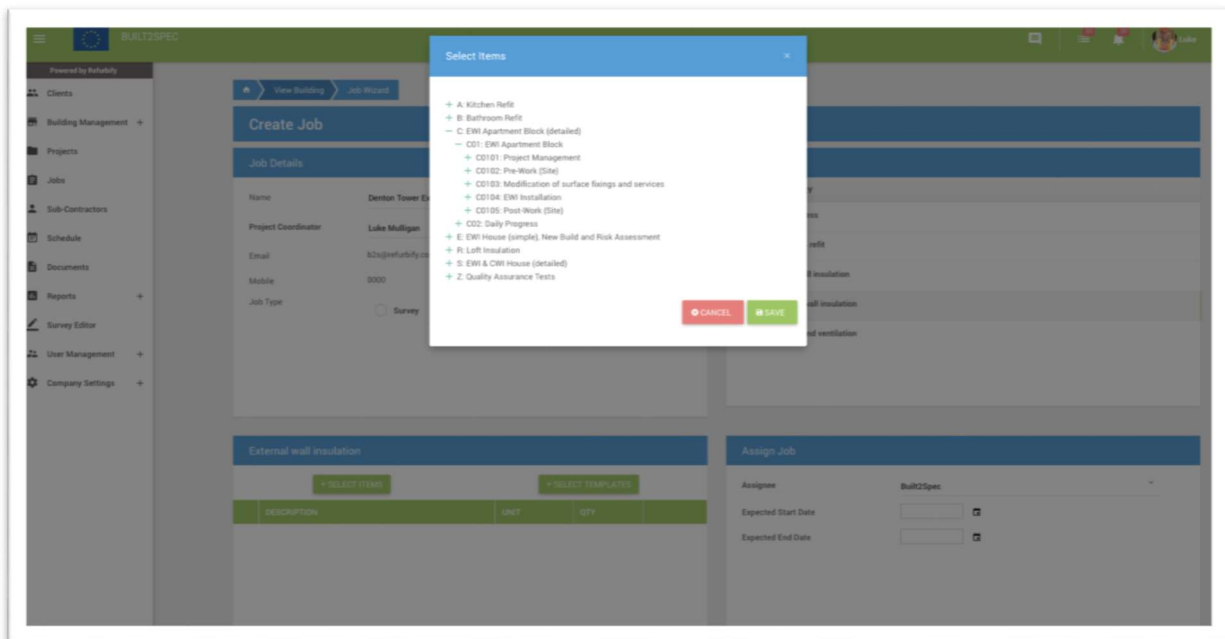
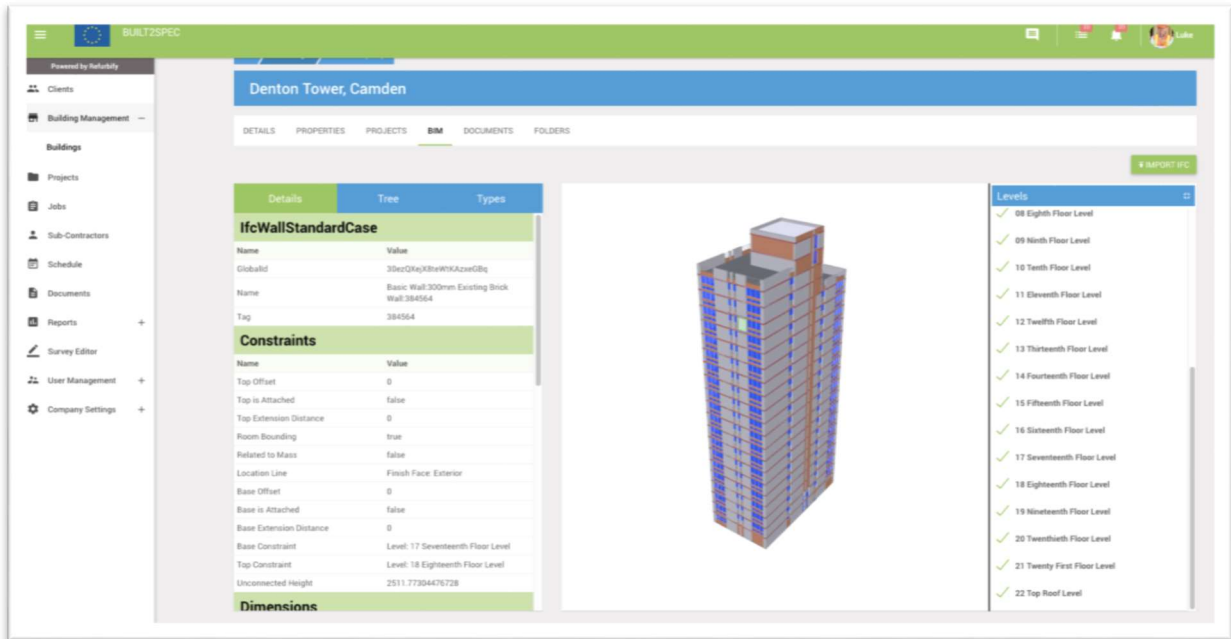
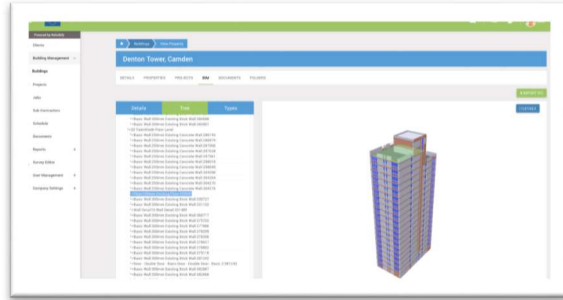
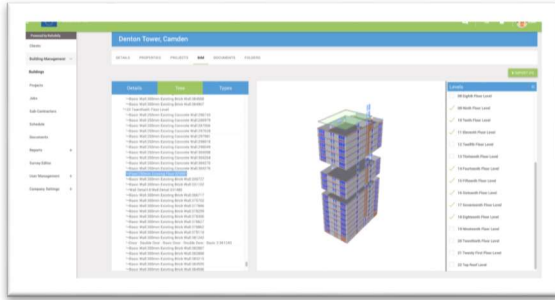
During the Built2Spec project the BIM Surfer tool was tested intensively using large datasets and many concurrent users. The project team concluded that the BIM Surfer tool is stable and ready for enterprise use.

Much appreciated features of the tool are the BIM specific features like:

- Performing BIM specific queries (using BIMserver as a backend)
- Filtering per storey, per classification, per object type, etc
- Advanced options for showing and hiding objects
- Options to change color and transparency of objects
- BIM specific camera controls (like fitting objects in view, fly to object(s), etc.)

This tool is also used in the regular version of VCMP (meaning the non-mobile version).





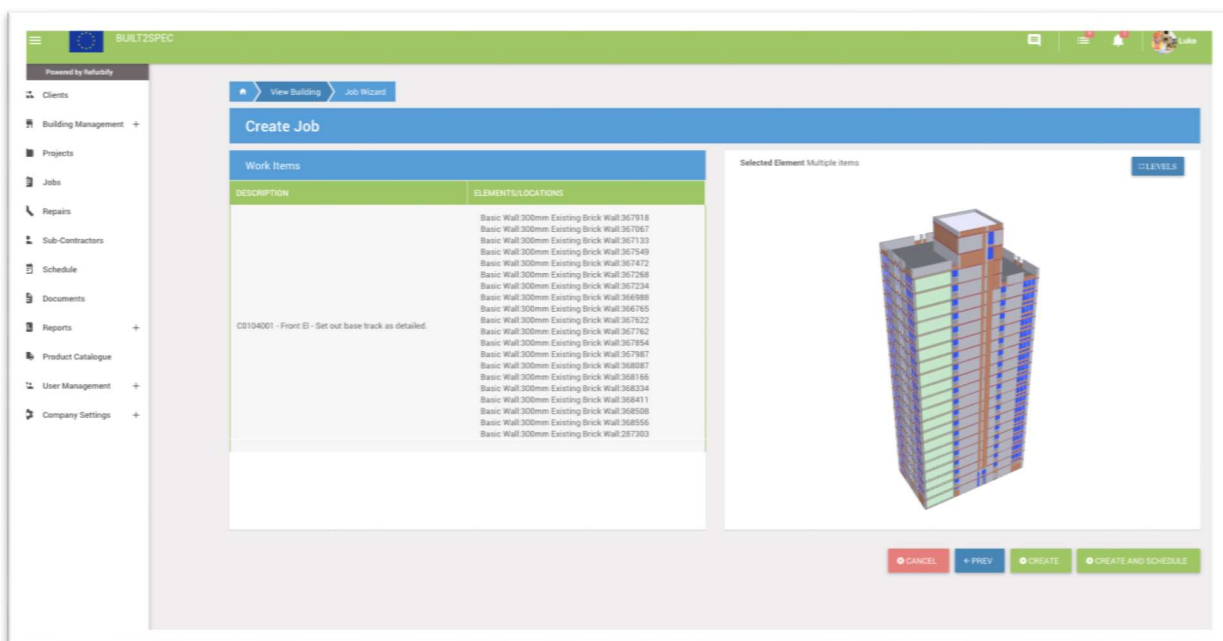
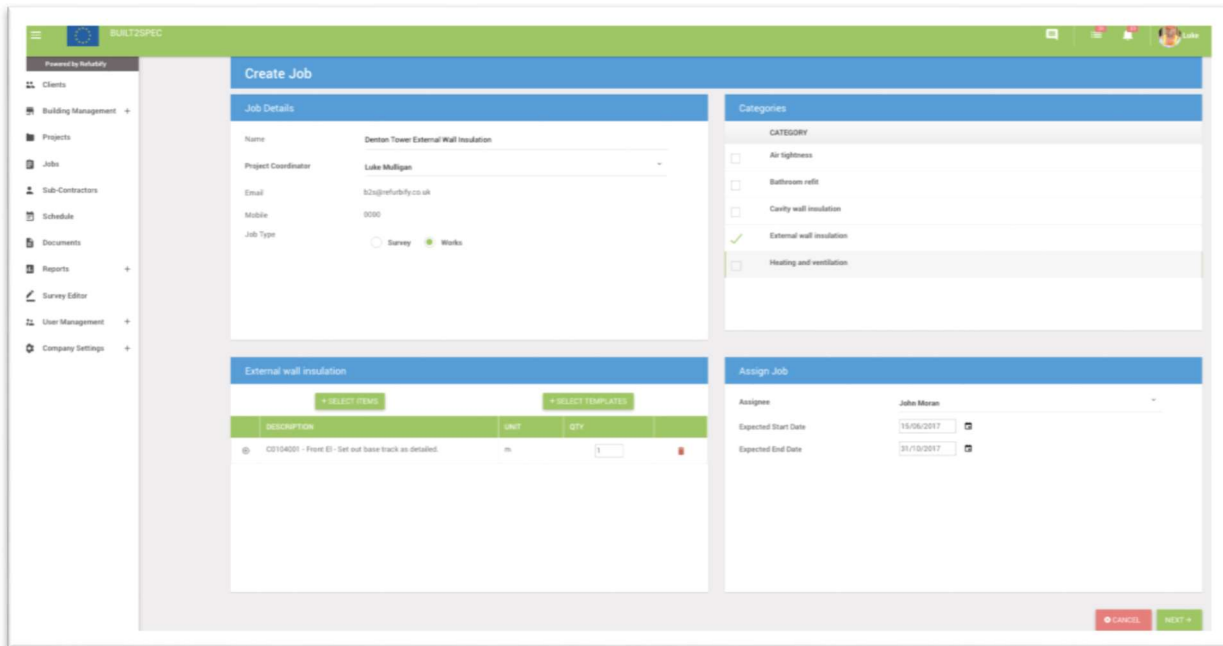


Figure 6.1 : Several screenshots of BIM functions integrated in the VCMP

For the Built2Spec project, the API of the BIM Surfer was redeveloped and extended with additional features. Specifically the selecting of objects is drastically improved. An example of the new API to select objects:

```

Selecting four objects:

bimSurfer.setSelection({ids: ["object3", "object2", "object4", "object6"], selected: true });

then querying which objects are selected:

bimSurfer.getSelection()

The result shows that those four objects are currently selected:

["object3", "object2", "object4", "object6"]

If we then deselect two objects, then query the selection again:

bimSurfer.setSelection({ids: ["object3", "object6"], selected: false });
bimSurfer.getSelection()

The result shows that only two objects are now selected:

["object2", "object4"]

Subscribing to selection updates:

bimSurfer.on("selection-changed",
  function() {
    var selected = bimSurfer.getSelection();
    console.log("selection = " + JSON.stringify(selected));
  });

```

Figure 6.2 : Example API of the new BIM Surfer WebGL viewer (source: BIM Surfer wiki)

Within Built2Spec these new features were necessary to link tasks of the VCMF/Refurbify to the specific objects in the IFC model.

The linking of non-BIM data to IFC objects did still face the GUID Integrity issues described in Chapter 6.

7 Integrate functionalities for BIM based quality checking

Within Built2Spec there are several different technologies, tools and systems that have to be connected. Recent developments in IT show promising results for the concept of ‘microservices’. The concept of “BIM Bots” harnesses the concept of microservices to the BIM and building industry.

A bot is an (online) system that triggers on an event. It performs a task with some form of “intelligence”. A BIM Bot can perform an analysis or simulation on your BIM data, it can enrich the model by adding detailed objects, splitting it for specific use, or perform any other task on your data.

7.1 BIM Bots

A user can configure a BIM Bot to run every time a certain event has taken place. By configuring multiple bots a ‘loosely coupled’ network of bots is being created for that user. The data are being sent between the bots, every time enriching the knowledge and/or data.

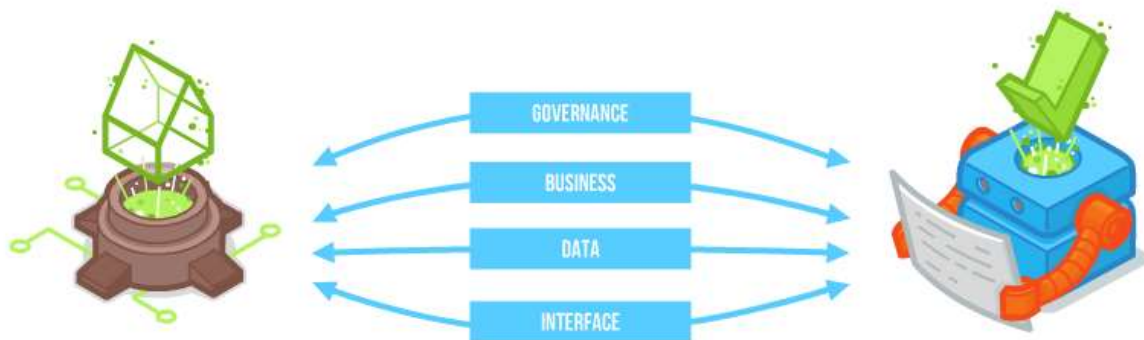


Figure 7.1 : The layers of connections between two online BIM systems

To link different Bots to each other, there need to be agreements on four layers: Governance, Business, Data and Technical Interfaces.

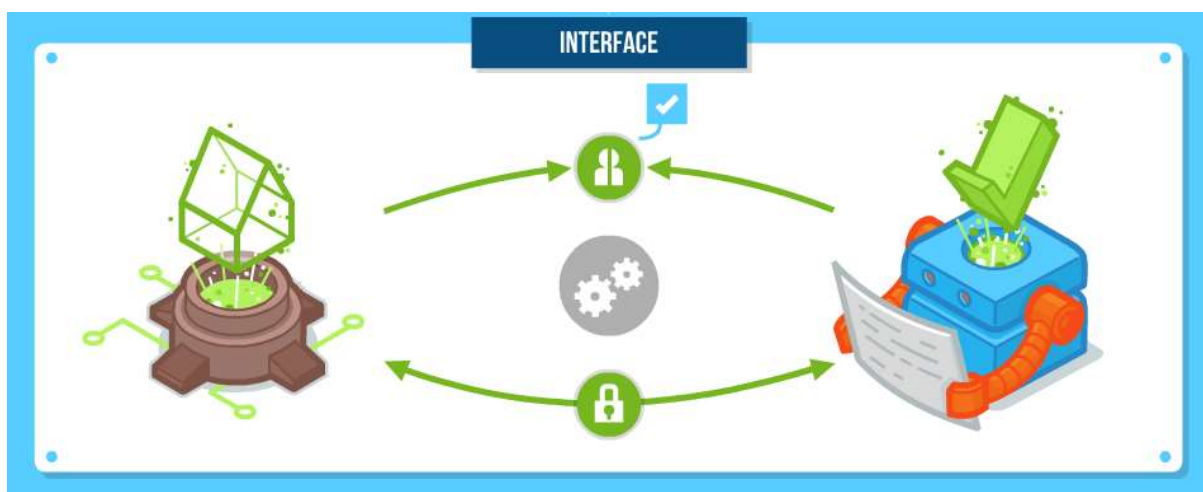


Figure 7.2 : The API of a BIM Bot

The technical interface between two bots is a standardized API. Technical calls like 'newRevision' or 'OAuthInterface.authorize' make it possible for two systems to connect to each other. By standardizing the interface, the connection and communication between two systems can be (semi-automated).

After an initial setup between two systems (configured by a user) one systems triggers the other system every time the configured event takes place. During the initial setup the user accounts belonging to the user on both systems are being linked so the server knows the users are actually the same person.

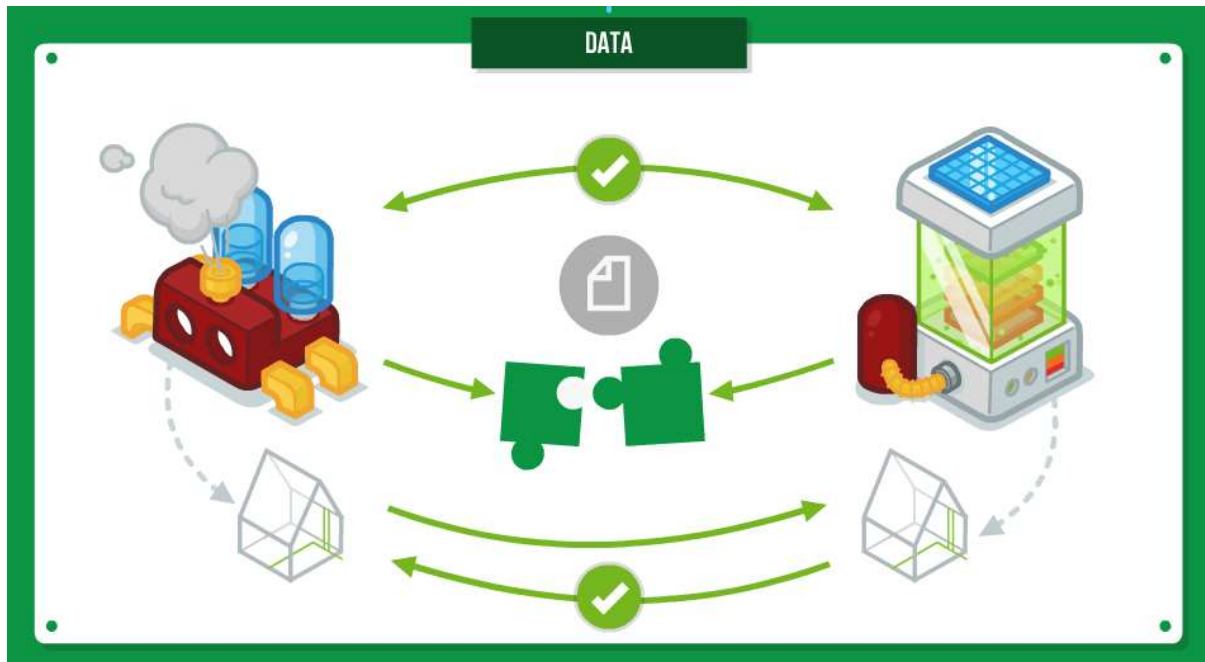


Figure 7.2 : The BIM Bot system is data agnostic

The concept of BIM Bots is data agnostic. During the setup both servers mitigate in which data standard they can provide or consume the data. This can be openBIM standards, but also (closed) proprietary data standards. The BIM Bots concept doesn't prescribe anything. The setup process takes place for data send to a BIM Bot and data that returns from it.

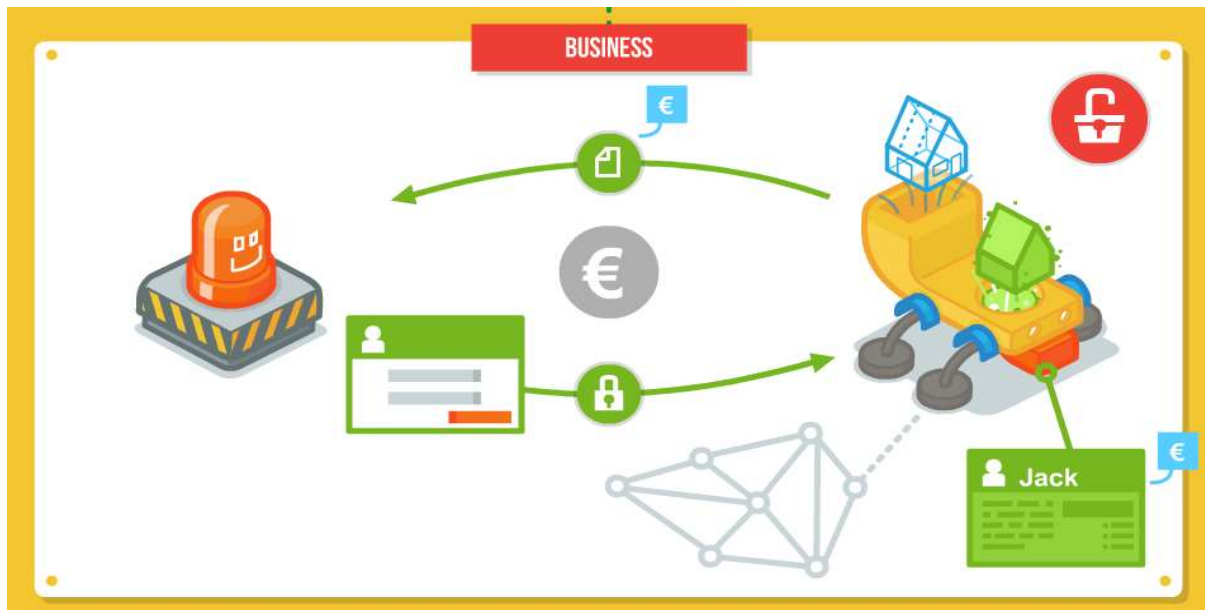


Figure 7.3 : BIM Bots can have their own business models

On the business level multiple options are available. A BIM Bot may ask for money for its usage. This can be a one-time fee, a subscription based on a fixed price per unit (e.g. per trigger, per result, per month, per year). The concept of ‘profiles’ is introduced here. A user might have several different profiles on a BIM Bot to fine tune the usage of the bot. This also gives the opportunity to edit settings of the behaviour of a BIM Bot. It is the responsibility of the BIM Bot to secure and encrypt this information.

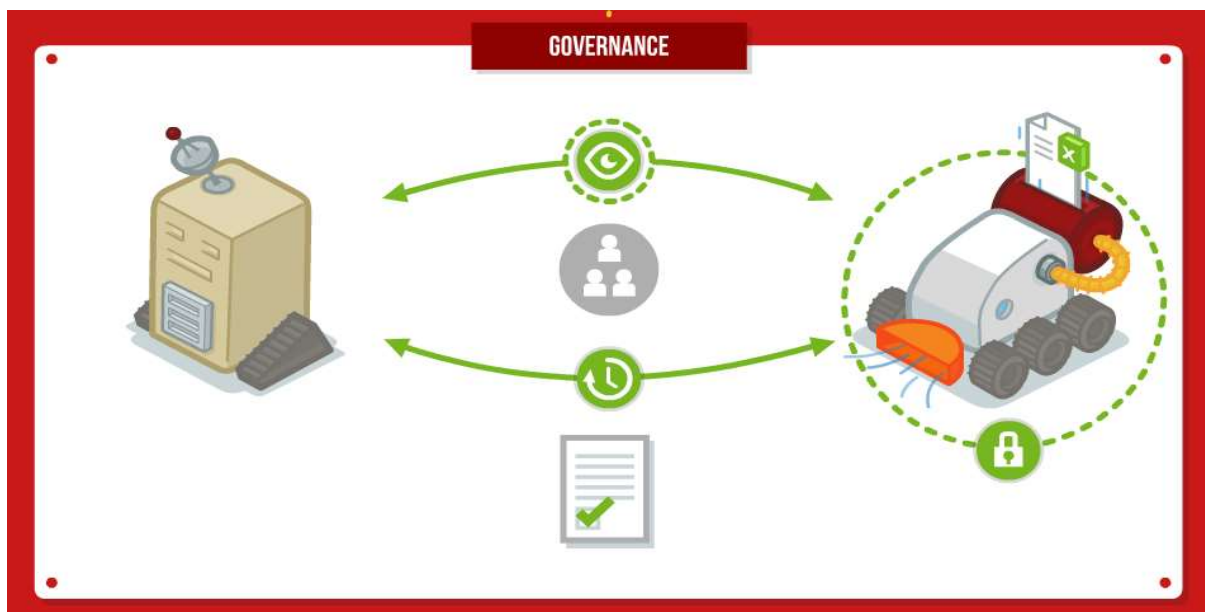


Figure 7.4 : Governance topics of BIM Bots

Every BIM Bot that is active has to provide terms and conditions of the service. In these terms and conditions statements are made about privacy of the users, security, reliability of the results, backups, encryption, etc. Additionally Service License Agreements can be provided to users to agree on more detailed issues like availability of the service, etc. There are example templates available for BIM Bot developers to use.

7.2 Example Built2Spec

The several tools and systems in Built2Spec all have a unique background, interface, governance, data structure and business model. The BIM Bots concept is an ideal concept to integrate the several tools and systems, while respecting their history and background.

An example of the Built2Spec BIM Bots setup for checking of the design BIM against the actual situation represented in a point cloud. The BIM Bots concept is placed in the '(automated) use case check' part of the example from chapter 4.4 (point cloud self-check).

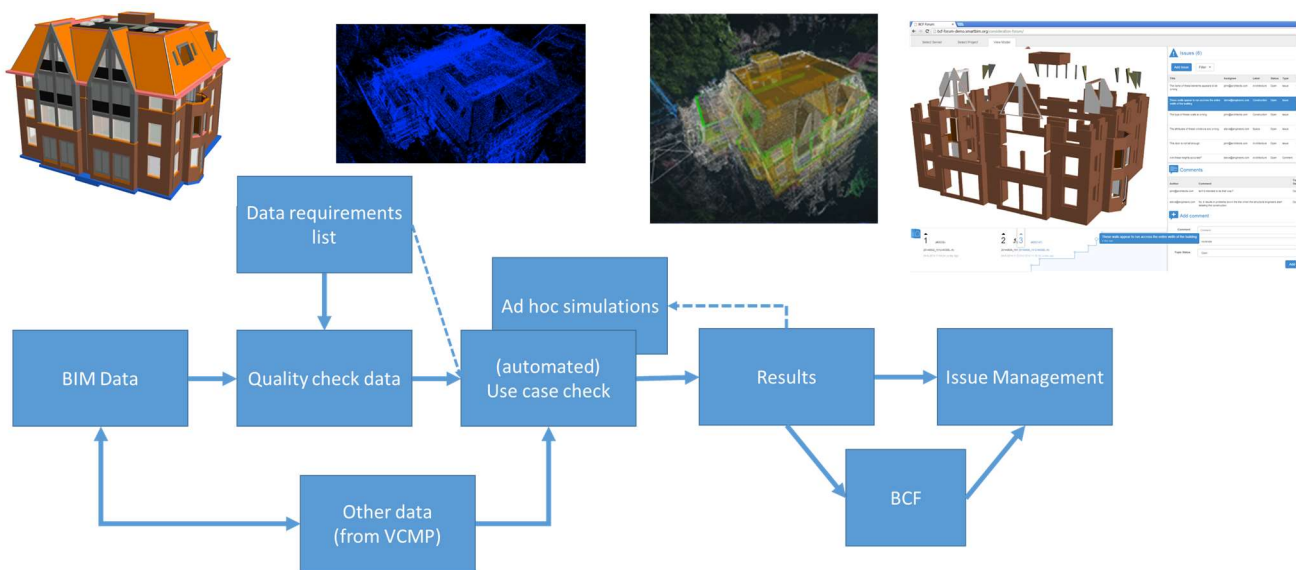


Figure 7.4 : the BIM Bot setup for the point cloud self-inspection.

The process based on BIM Bots can be explained as follows:

- The VCMP triggers the 'point cloud quality checker' from ETH to start checking.
- ETH triggers the BIMserver for the IFC data
- BIMserver delivers IFC data in the format ETH needs
- ETH gets the additional information from VCMP (in this case the list of maximum allowed deviations between the actual situation and the design)
- ETH checks point cloud data against BIM
- ETH creates a result and sends that to VCMP
- VCMP transforms the result to BCF
- VCMP triggers BIMserver with a new BCF
- BIMserver triggers users about an updated issue on the BIM

This can all be done in the background of the VCMP. This user does not leave the VCMP system.

In this setup, the power of all the separate tools is harnessed to the maximum ability, while giving the end-user the perception of an integrated system.

To make this technically possible, the BIM Bots API needed to be simplified and less tenuous. During the Built2Spec project, the BIM team worked on a new version of the BIM Bots API to fully harness the potential. As a result of these efforts, a journal paper has been submitted that details the BIM Bots concept in a more generic way.

8 Automated quality checking of ad-hoc design solutions versus predefined constraints

The BIM Bots concept described in the previous chapter provides technical capabilities to extend Built2Spec features in the near future.

Because the concept uses cloud systems, the computational power of the ecosystem can be triggered at any time. This means that simulations and quality checks can also be performed ad-hoc and on user request.

A big factor of failure costs on a construction site is in the long term effects of short term solutions. For example, when an issue arises on site, this is usually fixed as soon as possible and based on the experience and creativity of the site workers. During this action, the focus is on creating a situation where the issue that has risen will disappear as soon as possible, not always keeping the long term consequences in mind.

An example of this situation is when the foundation piles are not placed in the exact position, the site workers will create a practical solution so that the walls can be placed. Most of the time, the workers do not have an overview of other objects that need to be installed at a later building stage. Many times, their quick solution is blocking the MEP to be installed at a later stage.

By simulating the best solution in these situations, all perspectives, and all interest might be taken into account by the cloud system.

By evaluating and measuring the quality of the solutions, big data analyses can improve the chance of success in the long run.

In the future, this 'clash prevention' system might even be useful during the design phase of a project. That way designers improve the change of 'constructability' of their design by using a 'constructability' BIM Bot that is based on actual site workers input.

9 Issue management

Within the Built2Spec issues management of BIM data is a very important part of the innovation. In an IFC dataset, every product has a unique identifier, but this ID is not meant to be human readable. In practice, this makes communication about objects in an IFC dataset inefficient for users and there is a large margin for human errors. This is why the 'BIM Collaboration Format' (BCF) (<http://www.buildingsmart-tech.org/specifications/bcf-releases>) is introduced.

BCF is a data standard for BIM 'issues'. Technically BCF is a small standard. It holds not much more than a topic, description, a picture, IDs of the IFC objects with the issue, and coordinates of a camera viewpoint (how to best look at the issue). Each issue (which is called 'Topic' in BCF) can be assigned to a user, and classified with a priority, category, etc.

Both BCF and IFC are very much file based data exchange standards. Using BCF in a server environment can create issues in the workflow. This is elaborated by van Berlo and Krijnen (2014).

9.1 Issue management and BIM

The IFC file schema contains a central construct called the `IfcOwnerHistory`. It enables to relate, on a per-object basis, information about the `OwningUser`, or assignee of the object, and a state, a flag that pertains mostly to access rights. This in itself would be a crude way to model the status of objects, as the stakeholders in this project request a more fine-grained and extensible approach. On top of that, in the vast majority of IFC files, the owner history is not functioning as a central device to record revision history at all. In fact, in all publicly available IFC models in the `IfcOpenShell` repository of test models, which aggregates several sources of publicly available models, only 2 of the 122 contain multiple owner histories. Furthermore, within these two files, it does not associate information on responsible users to the products in the model.

Another alternative would be to model issues or comments as `IfcApproval` instances, which is the designated entity to model approval for certain parts of the design. However, the semantics of this entity are partly opposite to what the stakeholder wants to model, which would be a list of issues and associated responsible persons within the organization. Furthermore within the set of investigated test models not a single model features an `IfcApproval` instance, which would indicate that support for this construct is lacking in authoring applications.

A third approach would be to (ab)use the semantics of the `IfcPropertySet` construct, which is designed to associate arbitrary groups of key-value pairs with objects. These key-value pairs could be used to model the attributes of a comment, say a subject, assignee and date field. A nice consequence of this approach would be that many tools for viewing IFC files will already be able to show these user-defined property sets in their interface. However, the discovery of elements with comments would still be very hard: perhaps the only way would be to individually click all elements and manually inspect whether they have property sets that in fact describe an issue. In addition, easily adding comments is not something that is supported by this approach, since the viewing applications, mentioned above, do not support to append entity instances to the file. Therefore this approach is also not suitable to meet the requirements from the commercial partners.

9.2 BIM Collaboration Format

The BIM Collaboration Format defines an XML based schema, developed by several commercial software vendors, to exchange written commentary pertaining to certain IFC file. Such a BCF file, conventionally, would be transferred by regular file sharing means.

Technically, using the BCF schema, issues are related to a set of building elements from the associated model file. This relationship is of a textual nature. The IFC GlobalId (a globally unique identifier as a Base64 encoded string) attribute of the building elements is used to identify an element from the IFC model. As part of the VisualisationInfo of the BCF structure, this GlobalId is referenced as a Component to which the issue is related. The BCF schema validity is checked by means of regular XSD string validation. Hence, an issue that refers to a non-existent component can still constitute a valid BCF file. On the other hand, in a scenario where both the building and the issue would be modelled in the same IFC file, similar as outlined above, the validation of the comments would be part of the natural semantic validation of the IFC file. Such validation checks whether all entity references can be resolved to actual instances. Likewise, for example, in a relational database approach the validity of references could be enforced by conventional integrity checking using foreign keys in a relational database management system (RDBMS). In the BCF workflow this strict validation is lacking. This becomes an increasing problem once multiple revisions of the model are to be managed and, due to the file based nature of BCF, the association between the two would not necessarily always be clear.

A conventional BCF workflow describes a file based transmission of issues, say e-mail based, and does not propose means to centrally organize the communication that pertains to a model. This makes it difficult for stakeholders to analyse the entire body of issues and get a quick overview of the whole project. Note that these comments and issues contain valuable insights into how the design team functions and about the performance of families of products and suppliers. By not relying on a central agent to process the flow of issues, this valuable information cannot be easily uncovered. Furthermore, the amount of open issues gives an indication of the progress of the design project, but without a central registration of issues, this number is not accessible. In addition, central storage allows for easy notification to project members when a new issue is reported and enables it to be easily delegated to the appropriate person. Therefore, such a file based approach contradicts the desire of the commercial partners to offer a centralized project management interface that communicates the state of the project as a whole to project managers and clients.

Lastly, updating the attributes of an issue is not straightforward when issues only exist in bilateral conversations, as it would imply contacting all project members to which the issue has been sent and inform them about the changes manually. Furthermore, in general, this approach of simply storing issues and comments in files and sending them is prone to error as issues are not recorded and might slip through the cracks.

9.2.1 Conventional BCF based issue exchange

The conventional practice of 'issue management', when using the BCF methodology, is to create a BCF file that contains a list of issues. In a typical case, this is done by a delegation from the project team during a session on Friday. During these sessions the project manager has the lead. BIM analysis software like Solibri Model Viewer or Tekla BIMsight is used to find and analyse issues of BIM data, either being a single model or a set of models. The software generates a list of issues in

the BCF format. Every issue is now a 'BCF Topic' and the combination of several topics is put into a BCF file. The topics are discussed, filtered and the result is sent to all relevant project partners.

In subsequent sessions, a new BCF file is created with new topics. In the experiment, it turned out to be more efficient to create a new list of current issues, rather than evaluating the topics from the previous session and build from that list. Consequently, the topics from the previous session are deprecated. The drawback of this workflow is that no revision management on issues is possible. It is not possible to track which issues take a long time to solve and, more importantly, in this way, previously detected issues might slip through the cracks.

Occasionally, the project manager would notice a repeated emergence of the same (low priority) issues. In such an event, he would initiate a 'clean-up session' with the modelers, in which small issues, like marginal geometry errors, were to be fixed. This would stop lower priority issues from interfering with the general overview of issues of higher importance or urgency.

9.2.2 New process

In Built2Spec, the BCF issues are created by a BIM Bot and stored in the VCMP.

The new process therefore has been fairly unaltered, and included advantages not seen in the traditional approach. Most notably, because of the server based architecture underpinning the new workflow external parties, such as the client, could get an immediate overview of the project, simply by logging into the online system. All project members, including the project manager, have more options to filter issues based on priority, date, assignment, and of course 'status', because an important contribution from the centrally accessible listing of issues has been the ability for issues to be updated or even closed by team members, giving an up-to-date overview to the client and project manager.

Additionally, in this new process, the possibility exists for the client to create new issues from a web browser without the need to install complex software. This lowers the threshold for clients to get involved in the project in every stage of the design.

A more detailed assessment and conclusions will be made in D4.6 of the Built2Spec project.

10 Conclusions

There are several technologies and data requirements needed to make BIM technologically ready for self-inspection and quality check. This chapter summarizes the most important conclusions:

- 1) Using open BIM data standard IFC is recommended for BIM data exchange.
- 2) Modelling requirements need to be respected to ensure necessary data is available for each self-inspection and quality check (the details are listed in chapter 5).
- 3) Creating a monolithically central system to integrate all data and technologies is not recommended. This report concludes that the proposal from D1.4 to use linked services is the right approach. This report suggests to use BIM Bots for WP5.
- 4) There is still no standard BIM Query language to recommend in this project.
- 5) The GUID integrity issues when using IFC can be ignored using an appropriate workflow.
- 6) Issue management and results from self-inspection and quality checks is best communicated using the BIM Collaboration Format (BCF).
- 7) Using self-learning algorithms to predict and prevent constructability issues on site, is a promising technology.
- 8) Using IFC/BIM Data online can be facilitated by an online WebGL viewer (BIM Surfer).

References

- Amor, R., 2015, "Analysis of the Evolving IFC Schema", in: *Proceedings of the 32nd International Conference of CIB W78, Eindhoven, The Netherlands*, p. 39-48.
- Berlo, L.A.H.M. van., 2014. Global Perspectives on BIM: The Netherlands. Journal of The National Institute of Building Sciences, June.p. 32.
- Berlo, L. v., Derks, G., Pennavaire, C. & Bos, P., 2015. Collaborative Engineering with IFC: common practice in the Netherlands. Eindhoven, CIB, pp. 59-68.
- Berlo, L. v. et al., 2012. BIM Quickscan: Benchmark of BIM performance in the Netherlands. Beirut, CIB W78
- Berlo, L. v. & Krijnen, T., 2014. Using the BIM Collaboration Format in a server based workflow. *Procedia Environmental Sciences*, pp. 325-332.
- Berlo, L. A.H.M. van, & Natrop, M. (2015). BIM on the construction site: providing hidden information on task specific drawings. *Journal of Information Technology in Construction (ITcon)*, 20(7), 97-106.
- Built2Spec project, "Tools for the 21st century construction worksite," 2015. [Online]. Available: <http://built2spec-project.eu/>.
- ECTP, 2012 "Building up Infrastructure Networks of a Sustainable Europe. reFINE Initiative," European Construction Technology Platform.
- Eastman, C., Lee, J.-M., Jeong, Y.-S., and Lee, J.-K., 2009, "Automatic rule-based checking of building designs", *Automation in Construction* 18(8), p. 1011-1033.
- Eastman, C., Teicholz, P., Sacks, R. & Liston, K., 2011. BIM Handbook. A guide to Building Information Modeling for owners, managers, designers, engineers and contractors. 2nd ed. New Jersey: John Wiley & Sons.
- European Commission, 2010, "The Directive 2010/31/EU of the European Parliament and of the Council of 19 May 2010 on the energy performance of buildings," Official Journal of the European Union.
- Farias, T.M. de, A. Roxin, and C. Nicolle, 2015, "IfcWoD, Semantically Adapting IFC Model Relations into OWL Properties", in: *proceedings of the 32nd CIB W78 Conference on Information Technology in Construction, 2015, Eindhoven, Netherlands*.
- Jung, W. & Lee, G., 2015. The Status of BIM Adoption on Six Continents. *International Journal of Civil, Environmental, Structural, Construction and Architectural Engineering*, 9(5), pp. 497 - 501.
- Khemiani, L., 2005, "CORENET e-PlanCheck: Singapore 's Automated Code Checking System", *AECbytes, Building the Future* , 1–8.
- Krijnen, T. and J. Beetz, 2016, "An assessment of linked data ontologies for use in the AEC/FM domain", in: *Proceedings of the 33rd International Conference of CIB W78, Brisbane, Australia*.
- Krijnen, T. and M. Tamke, , 2015, "Assessing Implicit Knowledge in BIM Models with Machine Learning". in: *Modelling Behaviour*. Springer International Publishing. p. 397-406.
- Kriinen, T., & van Berlo, L. A. H. M. (2016). Methodologies for requirement checking on building models.
- Lee, Y.C., C.M. Eastman, and W. Solihin, 2016, "An ontology-based approach for developing data exchange requirements and model views of building information modeling", *Advanced Engineering Informatics* 30(3), p. 354-367.
- Mazairac, W. and J. Beetz, 2013, "BIMQL – An open query language for building information models", *Advanced Engineering Informatics* 27(4), p. 444-456.
- McLeod, R.J., 1995, *Management Information Systems: a Study of Computer-Based Information Systems*, 6th Edition, Prentice-Hall, Englewood Cliffs, NJ.
- Miller, G.A., 1995, "WordNet: a lexical database for English." *Communications of the ACM* 38.11, p. 39-41.
- Navigli, R. and S.P. Ponzetto, 2012, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network", *Artificial Intelligence* 193, p. 217-250
- Nawari, O., 2012, Automated Code Checking in BIM Environment, in: *Proceedings of 14th International Conference on Computing in Civil and Building Engineering*
- NBS, 2016. International BIM Report 2016, Newcastle upon Tyne: NBS.
- Nederveen, S. v. & Tolman, F., 1992. Modelling multiple views on buildings. *Automation in Construction*, 1(3), pp. 215-224.
- Palos, S., A. Kiviniemi, and J. Kuusisto, 2014, "Future perspectives on product data management in building information modeling", *Construction Innovation* 14(1), p. 52–68.

- Pauwels, P. and W. Terkaj, 2016, "EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology", *Automation in Construction* 63, p. 100-133
- Pauwels, P., W. Terkaj, T. Krijnen, and J. Beetz, 2015, "Coping with lists in the ifcOWL ontology", in: *Proceedings of the 22nd EG-ICE International Workshop, Eindhoven, The Netherlands*, p. 113-122.
- Pérez-Lombard, J. Ortiz, and C. Pout 2008, "A review on buildings energy consumption information," *Energy Build.*, vol. 40, no. 3, pp. 394–398.
- Smith, P., 2014. BIM Implementation – Global Strategies. *Procedia Engineering*, Volume 85, pp. 482-492.
- Solihin, W. and C. Eastman, 2015, "Classification of rules for automated BIM rule checking development", *Automation in Construction* 53, p, 69-82.
- Solihin, W., C. Eastman, Y.-C. Lee, 2015, "Toward robust and quantifiable automated IFC quality validation", *Advanced Engineering Informatics*, 29(3), p. 739-756
- Terkaj, W. and A. Šojić, 2015, "Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology", *Automation in Construction* 57, p. 188-201
- Zhang, C., J. Beetz, and M. Weise, 2014, "Model view checking: automated validation for IFC building models." *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM*.