

D 4.3: Framework

Framework for BIM, hardware and software data interoperability

FINAL Version



Deliverable Report: D4.3 , issue date on 11.30.2015

INSITER - Intuitive Self-Inspection Techniques using Augmented Reality for construction, refurbishment and maintenance of energy-efficient buildings made of prefabricated components.

This research project has received funding from the European Union's H2020 Framework Programme for research and innovation under Grant Agreement no 636063.

D 4.3: Framework

Framework for BIM, hardware and software data interoperability

Issue Date	November, 30 th , 2015
Produced by	RDF
Main author	Peter Bonsma
Co-authors	Jan Tulke, Jan-Derrick Braun, Richard Deighton
Version:	1.1
Reviewed by	José Luis Hernández, Pedro Martín Lerones, Carlos Bárcena Martín and Alberto Nafría Mundi
Approved by	Dr. Rizal Sebastian

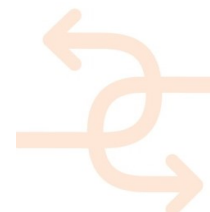
Colophon

Copyright © 2014 by INSITER consortium

Use of any knowledge, information or data contained in this document shall be at the user's sole risk. Neither the INSITER Consortium nor any of its members, their officers, employees or agents accept shall be liable or responsible, in negligence or otherwise, for any loss, damage or expense whatever sustained by any person as a result of the use, in any manner or form, of any knowledge, information or data contained in this document, or due to any inaccuracy, omission or error therein contained. If you notice information in this publication that you believe should be corrected or updated, please contact us. We shall try to remedy the problem.

The authors intended not to use any copyrighted material for the publication or, if not possible, to indicate the copyright of the respective object. The copyright for any material created by the authors is reserved. Any duplication or use of objects such as diagrams, sounds or texts in other electronic or printed publications is not permitted without the author's agreement.

This research project has received funding from the European union's H2020 Framework Programme for research and innovation under Grant agreement no 636063.



Executive summary

This report describes the Framework for BIM, hardware and software data inter-operability. The Framework is the architecture of relevant hardware, software and corresponding data formats and data flows within the INSITER project. This resulting Framework or any part of it can be applied for companies using INSITER results. Next to the abstract Framework a collection of tools and services is given as example materialization of the abstract Framework.

The framework as well as its materialization (specific applications filled in as examples or reference choices within the abstract structure) will grow during the project as more applications, data stores, converters and services will arise. This deliverable contains the basic abstract framework given by the requirements of the project INSITER and the status of the work done till now. The framework is requested to be developed early in the INSITER project. This means the requirements for the project are relatively 'high-level'. Further on in the project the requirements are expected to evolve into more detailed set of requirements. As Task 4.2 will be active for another 2 years the framework will be updated when needed. More detailed requirements from use-cases, selected hardware, testing of different applications and specifically the testing of BIM servers will improve and fine-tune both the abstract Framework as well as the materialization of the Framework with applications, application types and more detailed information flows.

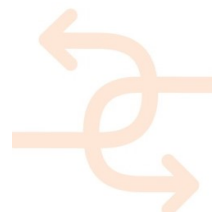
Deliverable D4.3 is of type 'other', it is not a document but a data schema (i.e. Framework). This document is however created accompanying the Framework to explain the ideas behind and the road towards the framework (D4.3) we created; this is the executive summary of the accompanying document of D4.3.

The Framework itself is described in chapter 5; it consists of the following items:

- The abstract structure of components and inter-connections.
- Applications, services and hardware used which are in one of the components.
- The data-formats and data-flows present.

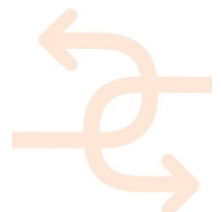
This document starts, after a short introduction, with explaining BIM and open standards. The next chapter focuses on the different sensor hardware INSITER is expected to support. The fourth chapter discusses the different software components, here we especially focus on software supporting open BIM standards, i.e. freely available IFC Viewers and available BIM / IFC server solutions. The fore-last chapter describes the actual Framework, i.e. the actual D4.3 result. This chapter will discuss the abstract Framework, where what software components can be embedded and the different data-formats and data-flows within the Framework. The document ends with the conclusions.

Deliverable D4.3 is expected to be directly the end-result of the Framework. However, we found the framework is depending on several decisions in the project that still need to be taken, like conclusion on use-cases and used hardware. If further developments within INSITER will change requirements we need to improve the Framework or update distribution of software types, data-formats and/or data-flows. The updated version of the Framework will be distributed within INSITER. The Framework is a public deliverable.



List of acronyms and abbreviations

- API Application Programming Interface
- AR Augmented Reality
- BIM Building Information Model
- CAD Computer Aided Design
- E57 ASTM E2807 standard for point-cloud data
- ICS Industrial Control System
- IFC Industry Foundation Classes (ISO 16739:2013)
- IPS Indoor Positioning System
- MEP Mechanical Electrical Plumbing
- OPC A group of open standards for Open Connectivity of Industrial Automation
- SCADA Supervisory Control and Data Acquisition
- XYZ A point-cloud format



Definitions

Augmented Reality

Augmented Reality is a technology that superimposes the computer generated virtual content over a live view of the world which not only contains the input from the 3D model but may also have sound, video, graphics or GPS data inputs enhancing our perception of real world situations. It has the potential to change how site managers and construction workers can interact and access to virtual information in real-time.

Building Information Model

BIM is a method for optimizing the design, execution and operation of building structures. The basis of BIM is formed by a 3D computer model which can be enhanced by adding further information, such as time, costs, utilization. It is not a software package but a method of working, collaborating, designing, managing, constructing and operating.

Clash Detection

Clash detection is a process for effectively identifying, inspecting and reporting interferences in a 3D project model. Instead of conducting clash detection only one-time for coordination of completed design work, it is used as a continuously on-going project audit and quality control process.

Industry Foundation Classes

Independent standard for describing the building models of various CAD systems. As well as the geometrical data, further properties of the building structures are depicted. The IFC are defined by the Industrial Alliance for Interoperability (IAI). The IFC permit the exchange across different software systems of construction and facility management data.

Laser Scanning

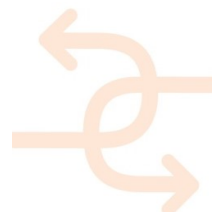
Laser Scanning describes a three-dimensional imaging method in which surface structures are computed by means of the pulse duration of a laser beam. This creates a digital point cloud with a defined number of points and their respective coordinates. Currently, scanning rates of up to 100,000 points per second with a precision of approx. 1 millimetre are possible.

Model Checking

Where Clash Detection is mainly referring to geometrical conflicts, Model checking describes the process of identifying conflicts of any kind within the 3D model. For example, conflicts in the model topology or conflicts related to the design (rule based conflicts).

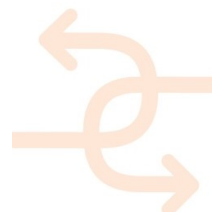
Indoor Positioning System

An indoor Positioning System (IPS) is a technique to determine locations inside a building using different technology than GPS (due to lack of unobstructed line of sight to GPS satellites). Until now there is no standard technology defined.

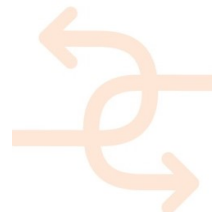


Contents

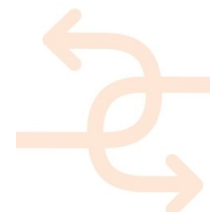
1. INTRODUCTION AND SCOPE	9
1.1 Requirements and Scope	9
1.2 BIM	9
1.3 Measurement Devices - Hardware	9
1.4 Applications and Services - Software	10
1.5 Architecture - Framework	10
1.6 Conclusions and Outlook	11
2. BIM	12
2.1 Definition of BIM	12
2.1.1 Building Information Modelling	12
2.1.2 Building Information Model	12
2.1.3 Building Information Management	12
2.2 Scope of BIM	12
2.3 Open BIM versus Closed BIM	12
2.3.1 Open API	13
2.3.2 Open Standards	13
2.3.3 Open Source	13
2.3.4 Closed BIM	13
2.3.5 Open BIM in INSITER	13
2.4 Data Exchange Formats	14
2.4.1 IFC Format	14
2.5 XYZ Format	14
2.6 E57 Format	15
2.7 SCADA Format	15
2.8 OPC Standards	15
3. MEASUREMENT DEVICES – HARDWARE	16
3.1 Thermal Camera's	16
3.2 3D Scanners	16
3.3 Sound Brush	17
3.4 Positioning Devices	17
3.5 Integration of Measurement Devices	17
4. APPLICATIONS AND SERVICES - SOFTWARE	18
4.1 BIM Client Applications	18
4.2 Open Standard BIM Viewers	18
4.2.1 Areddo	19



4.2.2	Bentley View V8i	20
4.2.3	BIM Vision 2.2	20
4.2.4	Constructivity Model Viewer / Editor	21
4.2.5	DDS Viewer	21
4.2.6	eveBIM	22
4.2.7	IFC++ Viewer	22
4.2.8	FZK Viewer	23
4.2.9	Tekla BIMsight	23
4.2.10	RDF IFC Viewer	24
4.2.11	Solibri Model Viewer	24
4.2.12	Xbim Xplorer	25
4.3	BIM Server Applications	25
4.3.1	BIM+	26
4.3.2	BIMCloud	26
4.3.3	BIM 360 Glue	27
4.3.4	Open Source BIM server	27
4.3.5	EDM Model Server	27
4.3.6	Adoddle	27
4.3.7	BIMSYNC	27
4.3.8	ILM	27
4.3.9	Think Project!	28
4.4	Self-Instruction and Self-Inspection Applications	28
4.5	BIM Services	28
5.	ARCHITECTURE - FRAMEWORK	30
<hr/>		
5.1	Framework - Components	30
5.1.1	Client Side – PC / Laptop / Tablet / Phone	30
5.1.2	Client Side – measurement devices	30
5.1.3	Server Side – applications	31
5.1.4	Server Side – services	31
5.1.5	Server Side - API	31
5.1.6	Server Side - Data Store	31
5.2	Framework – BIM	32
	32	
5.2.1	GUI	32
5.2.2	BIM Platform	32
5.2.3	Extended Functionality	32
5.2.4	INSITER Services	33
5.2.5	Middle Ware	33
5.3	Framework - File Types and Formats	33
5.4	Connecting Different Data	34



5.4.1	IFC referencing to external data sources	34
5.4.2	External data sources referencing to IFC	34
5.4.3	Independent data sources defining interlinks between data sources + IFC ('Linked Data')	35
5.4.4	External data embedded within IFC	35
6.	CONCLUSIONS AND OUTLOOK	36
6.1	Conclusions	36
6.2	Outlook	36
	APPENDIX 1 – SENSOR DATA IN IFC	37



1. Introduction and Scope

1.1 Requirements and Scope

The requirements of the framework are coming from the project description as defined in the 'Description of Action', the use-cases to be supported, the hardware and software selected. One of the requirements from INSITER is to use open BIM standards. Therefore we will discuss BIM, open BIM and open standards in the next Chapter. Chapter 3 will discuss the different types of hardware used in INSITER, as the actual selection of products is not finished at this moment the hardware is described in an abstract way. It is still point of discussion if sensor data of static, always present, sensors will be included; the focus will be on hardware that is placed temporarily as discussed in chapter 3.

The requirements of the use-cases are expected to become clearer during the project, if these updated requirements will influence the framework, the framework will be updated. The current framework is relatively abstract; we show the architecture, the possible applications type, data-formats and data-flows.

The main scope is to solve interoperability within INSITER. This will be done by defining the architecture, the connection between the components of the architecture and data-flows within the Framework.

1.2 BIM

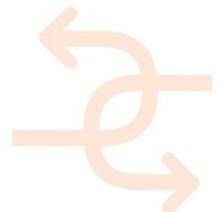
There are many different definitions of BIM. As BIM is an important subject within INSITER, a closer look at the different possible definitions if required. In Chapter 2 it is described what BIM is, what open BIM is and more in detail open API's, open standards and open source; these terms are often a source of confusion when talking about open BIM. Next to the definition of the BIM term itself, it is also needed to describe the scope of BIM.

This chapter will end with a small description of the most relevant open BIM standards. This will include IFC as well-known open BIM standard, but also standards for point clouds (the XYZ format and the E57 format) and SCADA as open standard for machine control.

1.3 Measurement Devices - Hardware

The measurement devices and sensors relevant within INSITER are the 'Hardware' used within the Framework. Chapter 3 will explain in more detail the Thermal Camera's, 3D Scanners, Sound Brush, Positioning devices and possible combinations of these devices.

For each type of measurement devices, the generated data are described. Generated data can be both raw data coming directly from the hardware or processed data. All data are processed, therefore it is hard to define a clear line between raw source data and processed data. The processing of data can be done within the hardware itself, or in a service or application converting the 'raw data'.



A more detailed description of the measurement devices can be found in the deliverables from WP2.

A final selection of devices is not clear at the moment, therefore a connection to firmware and middleware will focus on the definition of open standards for storing raw data and processed data from the devices. These standards can be found in chapter 2 about BIM.

1.4 Applications and Services - Software

From a user point-of-view it is easy to apply navigation and orientation on top of a 3D model for clear and correct understanding of the actual content that is presented. From technical point-of-view we are looking for a solution based on an open standard. As IFC is the most prominent open standard concerning BIM data including 3D representation a more detailed view on 11 freely available IFC Viewers on the market will be shown. The functionality of an IFC Viewer is expected to be an important component on client side within the Framework. Each IFC viewer is tested with the available IFC models from Task 4.1, see D4.1.

Similar to the IFC viewers on server side a BIM server is expected to be an important part of the data store component of the Framework. A set of different BIM servers is presented. Both the server solutions as well as the IFC viewer solution will have to handle not only IFC data but also data from the sensor data, i.e. the hardware and laser scan.

1.5 Architecture - Framework

In chapter 5 the Framework will be presented as given by the picture below. A description of the different components can be found there.

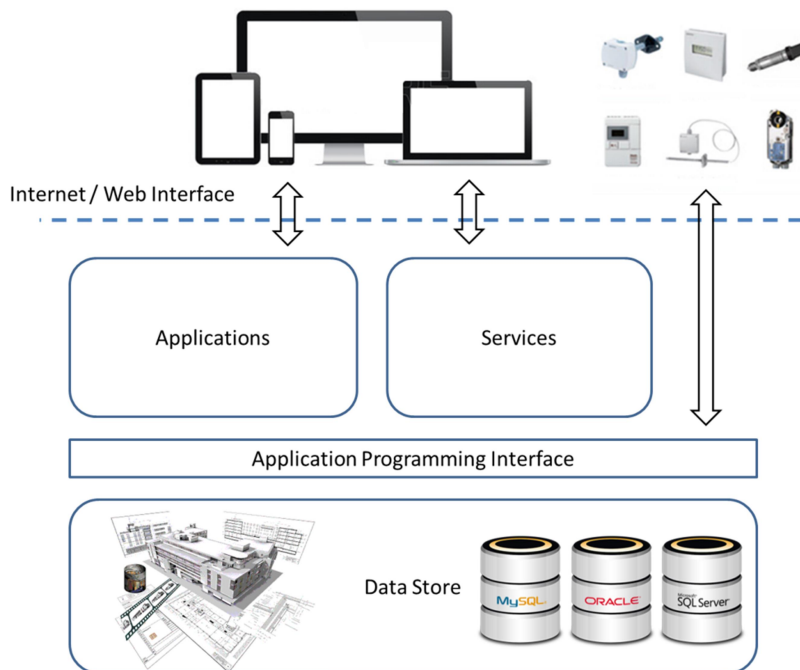


Figure 1, Framework.



Chapter 2 discusses the different open standards used in the data-store, created and used by the hardware and several software applications or services. It therefore forms the basics of the information flows within the Framework.

Chapter 3 discusses the hardware that generates data stored in the data-store, both the hardware and connection to the data-store can be found back in the Framework.

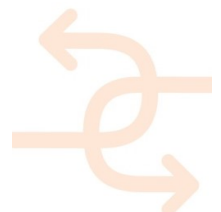
Chapter 4 discusses the software and goes into detail on an essential application as well as solutions for the data-store. The different applications reviewed are examples for possible solutions of the Framework.

Chapter 5 will discuss 4 different architectures of connecting the different data sources from both the CAD model and the data generated by the hardware. Technical details how these different data sources can be connected are placed in the Appendix

1.6 Conclusions and Outlook

The conclusions that can be taken from creating this Framework are presented in the last Chapter. Also the outlook based on relevant activities are described.

As requirements are expected to evolve during the project, especially as at the moment on creating the Framework the project is in an early stage, the Framework itself could evolve based on these updated requirements. Both Framework and more detailed selection of solutions for the different components will be kept up-to-date by Task 4.2.



2. BIM

2.1 Definition of BIM

BIM is a term often used within the Building & Construction industry. As much as the term BIM is used we find different definitions and interpretations of BIM. We find 3 major groups of definitions of BIM.

2.1.1 Building Information Modelling

This describes the process for generating and leveraging building data for design, construction and operation of the building during its lifecycle. BIM allows all stakeholders to have access to the same information at the same time through interoperability between technology platforms.

2.1.2 Building Information Model

This describes a digital representation, a virtual computer model of a project which holds selected structured data about the asset (design, quantity, time, cost, as built etc.). The results of D4.1 are actually the Building Information Models of the test cases / reference models from INSITER. Within chapter 4 we will use these models as input for testing and checking the software applications.

2.1.3 Building Information Management

This describes the organization of the business process by utilising the information in the digital prototype to effect the sharing of information over the entire lifecycle of an asset. The benefits include centralised and visual communication, early exploration of options, sustainability, efficient design, integration of disciplines, site control, as built documentation, etc. – effectively developing an asset lifecycle process and model from conception to final retirement.

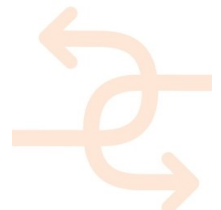
In general when talking about BIM without mentioning what is really meant it is safe to assume that a Building Information Model is meant. This is the definition that is meant to be considered within INSITER.

2.2 Scope of BIM

Given that BIM is defined as a Building Information Model, it is still not known what type of information is actually building information. In the context of this document the scope of BIM is defined in the broadest sense. This means not only building related object data like geometry, properties and relations to other objects, but also all information generated by hardware as described in chapter 3 is defined as part of BIM.

2.3 Open BIM versus Closed BIM

Having an Open BIM application can mean many different things. Here we describe the basic levels of open BIM within different applications.



2.3.1 Open API

Almost every software application in the Building & Construction sector has its own format to store its project/user specific information in a way it can reopen the same state at a later point. The definition and structure of this information is often dedicated to that specific application.

In case the storage and retrieval of internal data is not done in an open format, the program can still be open in defining an API (Application Programming Interface). The definition of an API is normally open when the application vendor wants third parties to make use of the API.

An Open API can be used for writing/updating information and/or for data retrieval. Open API's are often published with examples and many 'closed' applications have an Open API that they also use themselves to create support for different import/export formats.

2.3.2 Open Standards

If an application has support for an open standard this means that an application is able to import or export project specific information towards a structure that is described in a way that everybody is able to access, i.e. the specification is published. Open standards do not have to be defined by official organizations; however this is often the case to make use, acceptance and certification easier.

The support of open standards does not mean for importing that the application understands and parses every part of the content described. Neither does it mean for exporting that all relevant information is transferred to the file according to the open standard. Certification of an application however ensures that in most cases a specific subset of information is parsed correctly (in case of import certification) or a specific subset if exported correctly (in case of export certification).

2.3.3 Open Source

Often the terms open standards and open source are confused. Where open standards talk about an open specification of a data format, the term open source means that the source code of an application is available for every-body.

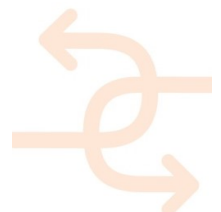
Open source does not mean by definition non-commercial. Many non-commercial projects creating software result in open source results, but the majority of serious open source projects have a commercial company behind.

2.3.4 Closed BIM

A closed BIM application is normally an application mainly uses propriatry data formats and does not fully support data exchange via open standards. This means that when we talk about open BIM applications they are not by definition open source.or non-commercial.

2.3.5 Open BIM in INSITER

Within INSITER we will focus on applications that support Open BIM in a sense that they support open BIM standards. For the Building & Construction the most prominent open standard is IFC, however for Augmented Reality, point clouds



and other sensor data there are many other open standards. We will explain a few of these open standards.

The choice for open BIM decreases vendor dependencies and reusability of INSITER results. Within the Building & Construction industry there is a well accepted and good available open standard that focusses on covering the complete lifecycle of a building, i.e. the open standard IFC. Next to IFC there are several other open standards like CIS/2 and gbXML, however these other standards focus on a small subset of the information relevant within INSITER, i.e. CIS/2 focusses mainly on steel constructions and gbXML mainly focusses on energy related information.

2.4 Data Exchange Formats

The following open exchange formats, i.e. open standards, are identified as most probably relevant for INSITER and therefore relevant for the Framework to support.

2.4.1 IFC Format

The IFC format was first created in 1994 as a 'work' dedicated to the Building & Construction of STEP developments dating back to 1985. In 1995 it became a vendor independent standard and had many releases, IFC 151 and IFC 20LF (Long Form) were popular releases for the academic world. Since IFC 2x3 released back in 2006 (and later IFC 2x3 Final and IFC 2x3 TC1) it was also becoming more popular for use in real life projects. Nowadays IFC 2x3 TC1 is still the most popular version although for INSITER the new version IFC4 (formally known as IFC 2x4) has some important improvements.

IFC is meant to be used by all the disciplines in the Building & Construction industry and the only native open standard supporting so many different disciplines. All major CAD vendors and a wide variety of other applications have support for IFC. Most of the applications supporting IFC are not certified, although most CAD systems with IFC support have a certification from buildingSMART. IFC carries object based view of the model, including geometry in 3D (and partly 2D) as well as properties and interrelationships between objects. It is a well thought through standard with relatively high complexity for software vendors to support. IFC supports also schedule data, quantities and many other construction related data.

2.5 XYZ Format

One of the open data formats for storing and exchanging point cloud data is the XYZ format. Even more than its name suggest the XYZ format is a simple small format of purely raw data of vertices (i.e. points) represented as XYZ coordinates. There is an optional header containing meta information about the information in the file.

An XYZ format file can be either a mesh or a scan. In case of a mesh the list of points is representing a list of lists in a way a 2D grid is formed. This way a surface can be created instead of just a set of points.

A scan has the ability to store slightly more detailed information about the points. In some cases scans can even represent solids, what makes them available for geometry operations. Typically colors and photo's cannot be stored in this format.



2.6 E57 Format

The E57 format also is able to store point clouds but is a more complex format and can store also other information from 3D scanners like images. The file format is specified by the ASTM, an international standards organization, and it is documented in the ASTM E2807 standard.

Next to the standard a 'reference implementation' is created to make use of the standards more attractive and easy. His reference implementation is called libE57 and is written in C++ and sources are available. The libE57 application contains an API that can be used by parties that like to import or export files in E57 format.

2.7 SCADA Format

SCADA (supervisory control and data acquisition) is a system that operates with coded signals over communication channels so as to provide control of remote equipment (using typically one communication channel per remote station). The control system may be combined with a data acquisition system by adding the use of coded signals over communication channels to acquire information about the status of the remote equipment for display or for recording functions. It is a type of industrial control system (ICS) (source: wikipedia).

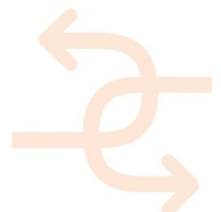
As an process control system SCADA can be important for the sensor hardware devices that will be used within INSITER.

2.8 OPC Standards

The OPC (formally known as Object Linking and Embedding for Process Control) standards are a group of open standards for open connectivity of industrial automation. The OPC Foundation is an industry consortium that creates and maintains these standards. This includes certification and a set of applications and libraries that can be used by companies implemented one of more of these standards. The OPC started in 1994 and released the first OPC standards in 1996. Three interesting groups of standards:

- OPC Data Access; this group of standards provides specifications for communicating real-time data from data acquisition devices such as PLCs to display and interface devices like Human-Machine Interfaces. The specifications focus on the continuous communication of data
- OPC Data eXchange ; this group of standards addresses server to server communications across industrial networks. The standards also address remote configuration, diagnostics, monitoring and management communications.
- OPC Historical Data Access; standards for communicating stored data

The OPC Foundation has a well established Certification Process. The OPC Foundation calls this the OPC Enhanced Certification Program (source: wikipedia).



3. Measurement Devices – Hardware

3.1 Thermal Camera's

A thermographic camera (also called an infrared camera or thermal imaging camera) is a device that forms an image using infrared radiation, similar to a common camera that forms an image using visible light. Instead of the 400–700 nanometer range of the visible light camera, infrared cameras operate in wavelengths as long as 14,000 nm (14 μm). Their use is called thermography (source: Wikipedia).

Within INSITER the thermal camera's expected to be used for energy loss, finding thermal leaks, etc. From the Framework point-of-view of importance is where the cameras are used, what kind of information the cameras generate, how they are triggered and when, where and how the information needs to be processed.

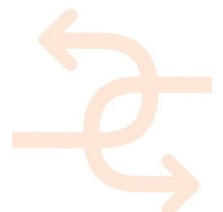
Thermal cameras will normally generate raw data as a 2D grid with a value for each point in the grid representing the thermal value of this point in the grid. With help of processing this raw data a 2D image can be created that represents the thermal behavior of an area (the surface). In case also a 'normal' camera is included the results could be shown directly in context with help of Augmented Reality.

3.2 3D Scanners

A 3D scanner is a device that analyses a real-world object or environment to collect data on its shape and possibly its appearance (e.g. colour). The collected data can then be used to construct digital three-dimensional models.

Many different technologies can be used to build these 3D-scanning devices; each technology comes with its own limitations, advantages and costs. Many limitations in the kind of objects that can be digitized are still present, for example, optical technologies encounter many difficulties with shiny, mirroring or transparent objects. For example, industrial computed tomography scanning can be used to construct digital 3D models, applying non-destructive testing (source: Wikipedia).

3D scanners create a set of 3D points (a 'point cloud') as raw data. A limited amount of such data could be directly visualized in CAD systems in case a dedicated connection is available, as the amounts of data are enormous for visualizing all data dedicated point-cloud software can be used. The raw data can also be exported to a file; well-known open formats are the so-called XYZ format and the E57 format. A more detailed description of these formats can be found in the previous chapter about BIM. Processing point clouds can result in an optionally closed set of faces. Faces are much easier to handle for CAD systems than just raw point cloud data, especially if non-significant points are removed.



3.3 Sound Brush

A Sound Brush is either a sound pressure microphone or a 3D intensity sensor. It measures the intensity of sound pressure from one or more directions. The results can be visualized in 3D where intensity is shown in colors and direction by lines or stored as raw data depending on the software used to process the data originating from the sensor.

The raw data and processed data are similar to data coming from thermal cameras.

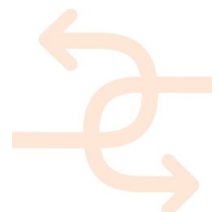
3.4 Positioning Devices

Positioning Devices can define the position (and rotation). This position can either be global, i.e. a position somewhere in the world or local compared to a beacon placed at a location relatively close to the device. Every positioning device has certain accuracy; the accuracy is very relevant for the purpose the device is used for.

3.5 Integration of Measurement Devices

Integration of Thermal Camera's, 3D scanners and/or a Sound Brush with positioning devices can establish the connection between BIM and the raw data more easily and possibly in an automated way. As there are a lot of different devices and connection of several measurement devices is also a challenge on hardware level the expected configurations of combinations of devices is expected to come from the hardware developers and universities within INSITER. The basic raw data and processed data will not change hugely, however as more information is known about when and where the data is scanned/created a more automated conversion and connection with BIM data can be expected.

Combined measurement devices will deliver combined sets of data. Integration of the hardware is complex; however the generated data itself does not expect to change. Processing the combined data is something that is needed to be created.



4. Applications and Services - Software

4.1 BIM Client Applications

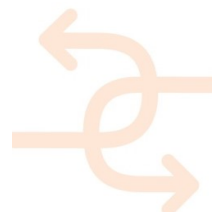
From a user point-of-view it is easy to apply navigation and orientation on top of a 3D model for clear and correct understanding of the actual content that is presented. From a technical point-of-view we are looking for a solution based on an open standard. As IFC is the most prominent open standard concerning BIM data including 3D representation a more detailed view on 11 free available IFC Viewers found will be shown. The functionality of an IFC Viewer is expected to be an important part of the application component within the Framework. With many viewers freely available for use we did a small research on their performance and features given the models that are currently created within Task 4.1. These models are Labtest room at UNIVPM and the mockup factory at Dragados. We will use the same models to have a more detailed look at server solutions that support IFC. As access to server solutions is more complicated and many of them are not free to use we will focus on two applications that gave us in the context of INSITER access to their development environment and BIM solution/database.

The existence of BIM viewer applications as well as the existence and functionality of BIM server solutions is of importance for the framework. INSITER is too small and next to that it seems useless to create BIM applications completely from scratch. Therefore we need to depend on existing developments. To understand better what developments are present and if and how they can be used, we looked more closely to applications that can visualize BIM models and at solutions that are able to process, store and handle BIM data at an abstract and extendible level. This means we looked at IFC Viewers and BIM server solutions with a focus on IFC.

4.2 Open Standard BIM Viewers

Although viewing IFC files is not very straightforward, still many IFC viewers can be found. Most of the viewers are free to use and some of them have source code available. This enables dedicated extensions for clients and/or for the INSITER project.

Visualization of IFC is more complicated, then support for CityGML, VRML, X3D etc. as part of the initial design intentions are kept within the file. For example openings or recesses within walls are defined separately from the geometry of the solid wall itself. This means that, to visualize the wall correctly, the viewer has to calculate the resulting geometry. The calculation for subtraction of openings from solids is called a Boolean Operation and, unless the individual cases are very restricted, such calculations are complicated. Support for this therefore requires availability of a so called geometry kernel. In several cases external geometry kernels are used as there is only one mature open source solution. i.e. Open CasCade and Open Scene Graph, we see this component coming back in at least three of the mentioned viewers below. Other viewers use a commercial Geometry Kernel or they have their own development. Even in our small test cases coming from task 4.1 we can see that some viewers incorrectly calculate the resulting visualization.



BIM Viewers found in alphabetical order:

- Areddo
- Bentley View V8i
- BIM Vision 2.2
- Constructivity Model Viewer / Editor
- DDS Viewer
- eveBIM
- IFC++ Viewer
- FZK Viewer
- Tekla BIMsight
- RDF IFC Viewer
- Solibri Model Viewer
- Xbim Xplorer

4.2.1 Areddo

Download location: <http://www.areddo.nl/>.

This is a free viewer offered by the Dutch CAD system developer Arkey. The interface is in Dutch. Typical functionality:

- Small and fast
- No installation needed
- Supports Clash detection and loading of several models at once

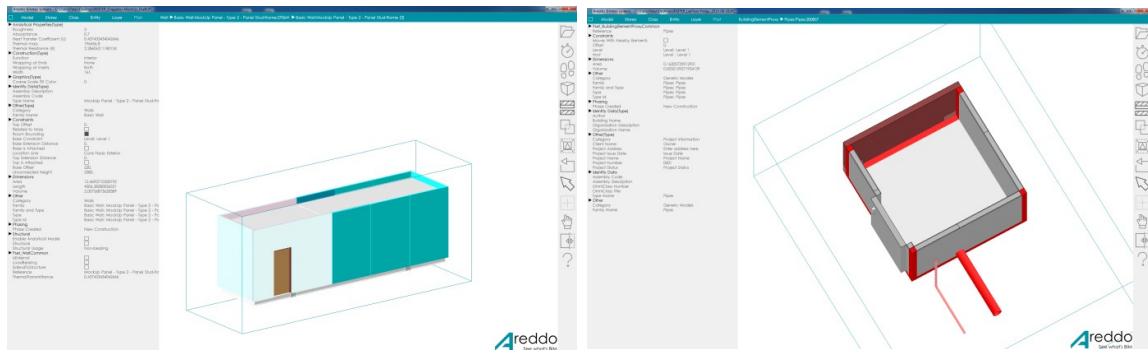
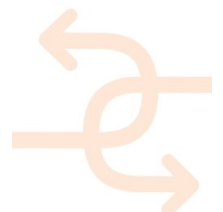


Figure 2, screenshots of Areddo.

Other applications from the same vendor:

- Arkey, a 2D CAD application with IFC support
- Adomi, a 3D CAD application with IFC support

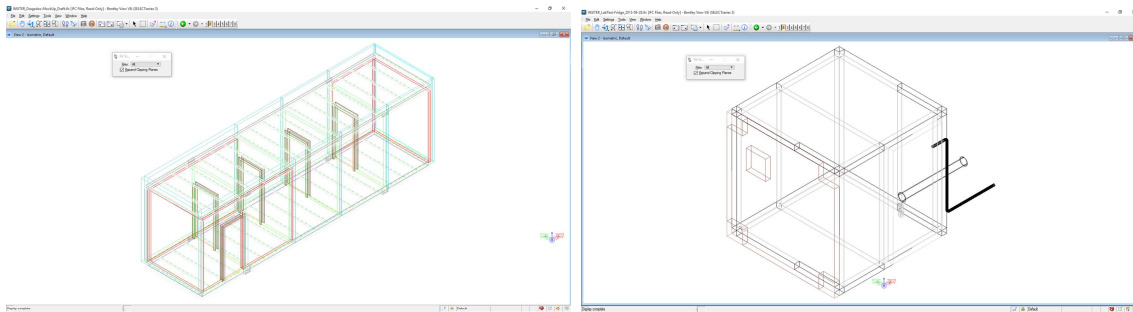


4.2.2 Bentley View V8i

Download location: <http://www.bentley.com/en-US/Products/Bentley+View/>.

Bentley is a large software company and the developer of several CAD systems, among them Microstation, STAAD and GenerativeComponents. Typical functionality:

- Several file formats including LandXML, DWG and point clouds
- Section views



Other applications from the same vendor:

- Microstation, CAD system for engineering and architectural projects
- ProjectWise, a project collaboration and content management platform
- AssetWise, to combine 2D/3D intelligent infrastructure models and point clouds with engineering information and asset performance management,

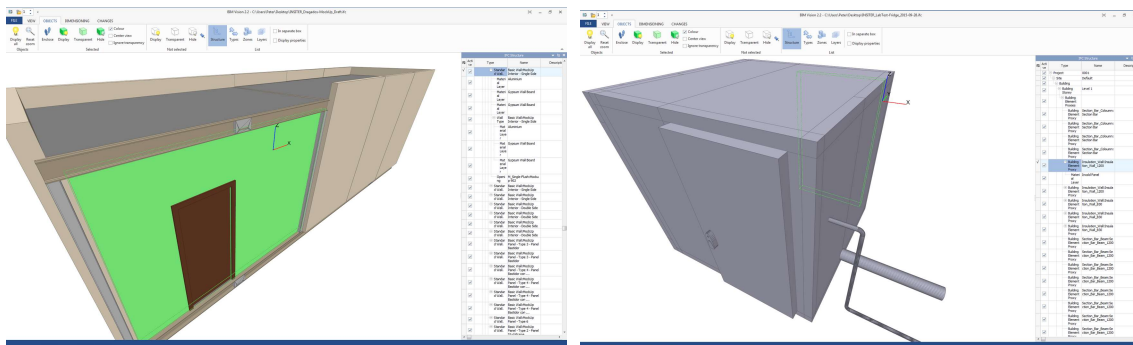
Figure 3, screenshots of Bentley View V8i.

4.2.3 BIM Vision 2.2

Download location: <http://www.bimvision.eu/>.

BIM Vision is developed by DataComp, a Polish software company also developing metriCAD. Typical functionality:

- Comparing models
- Projection views and 2D views
- API enabling companies to write plug-ins



Other applications from the same vendor:

- Geomagic Design - CAD/CAM application
- M-Files - document management system
- ZUZIA / BIMestiMate - cost estimating

Figure 4, screenshots of BIM Vision 2.2.



4.2.4 Constructivity Model Viewer / Editor

Download location: <http://www.constructivity.com/>.

Constructivity is a software company that offers a software suite around the IFC standard. As member of MSG the software company plays an important role in the publishing of the standard documentation via the so called ifcDOC tool they developed. Typical functionality:

- Support for the latest releases IFC and ifcXML in combination with exotic subjects like textures and Splines/NURBS
- Shape editing

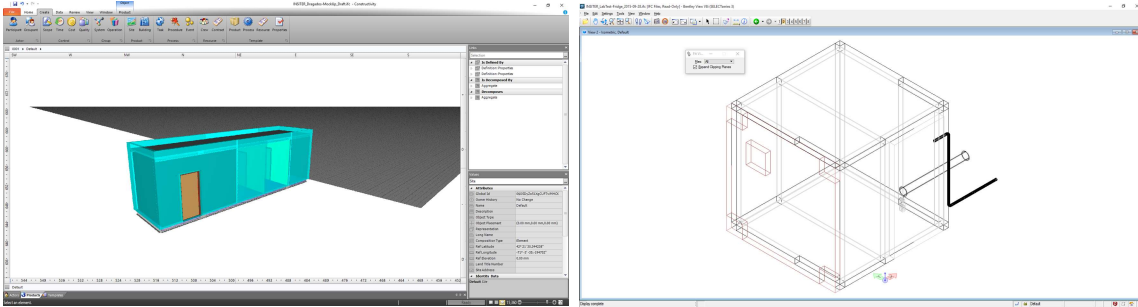


Figure 5, screenshots of Constructivity Model Viewer.

Other applications from the same vendor:

- Constructivity Model Server, model server solution

4.2.5 DDS Viewer

Download location: <http://www.dds-cad.net/downloads/dds-cad-viewer/>.

Data Design System is a software developer of the DDS-CAD software that supports several MEP modules. They are busy with IFC support for a long time. Recently the company is taken over by Nemetschek. Typical Functionality:

- Support for other file formats including BCF, gbXML, DWG, DXF

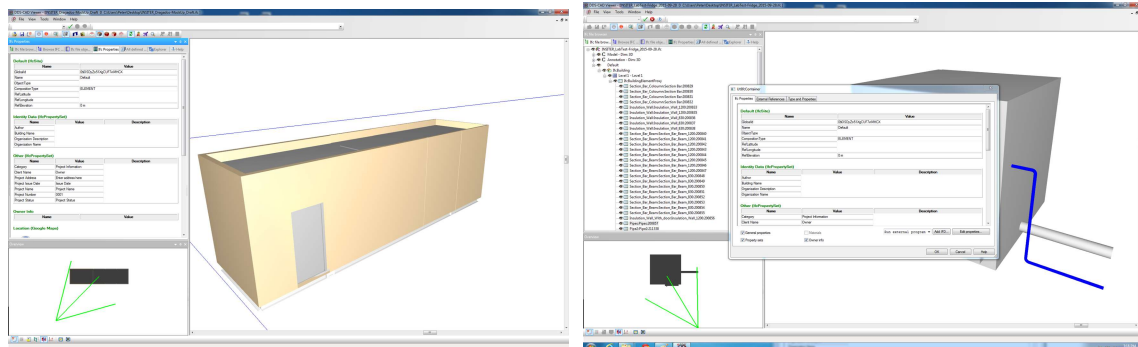
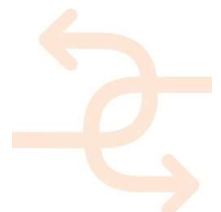


Figure 6, screenshots of DDS Viewer.

Other applications from the same vendor:

- DDS-CAD Mechanical, Electrical, Plumbing, Building, PV

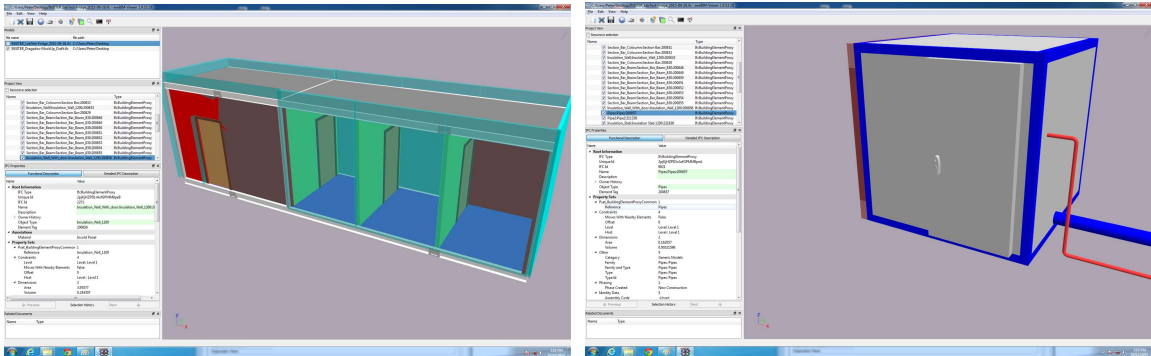


4.2.6 eveBIM

Download location: <http://salle-immersive.cstb.fr> (package location currently unknown).

CSTB is a large research center in France with a long history on IFC. The eveBIM viewer is an internal development of CSTB that is not that well known in the IFC community. Typical Functionality:

- Many connections to dedicated services
- Part of the sources are available



As CSTB is a Research organization they are not a typical product oriented organization. However they develop several products and work on extensions for IFC in the Civil area. Figure 7, screenshots of eveBIM.

4.2.7 IFC++ Viewer

Download location: <http://www.ifcplusplus.com/download.php>.

IfcPlusPlus is an open source C++ class model, as well as a reader and writer for IFC files in STEP format. A simple viewer application is made available based on Qt and OpenSceneGraph. Typical functionality:

- all source code available

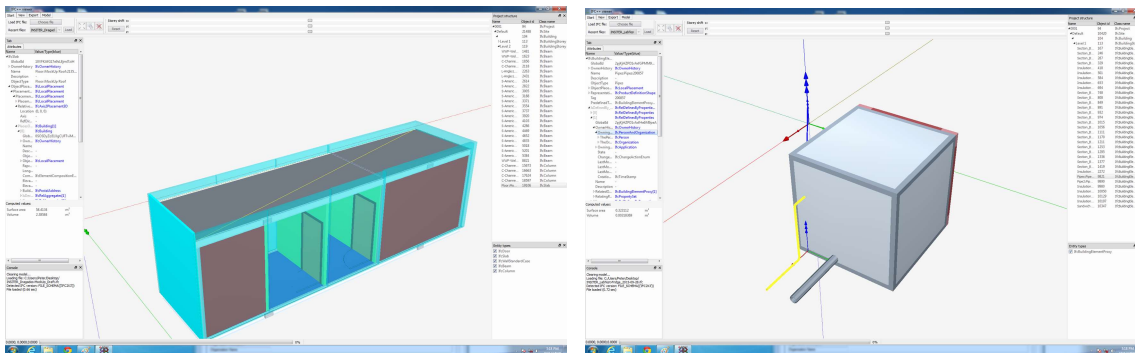
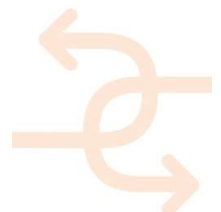


Figure 8, screenshots of IFC++ Viewer.

Other applications from the same vendor:

- Open Source C++ Class model + STEP import/export module,



4.2.8 FZK Viewer

Download location: <http://www.iai.fzk.de/www-extern/index.php?id=2315&L=1>.

Karlsruher Institute fur Technology (KIT) is an important party within the testing and certification of IFC files. The viewer they deliver enables detailed and expressive feedback on issues within IFC files. The same technology is used online as IFC testing server, i.e. the official test and certification server for IFC (<http://gtds.buildingsmart.org/>). Typical functionality:

- Especially well-suited for testing the quality and correctness of an IFC file
- Enables integration of GIS (CityGML) and BIM

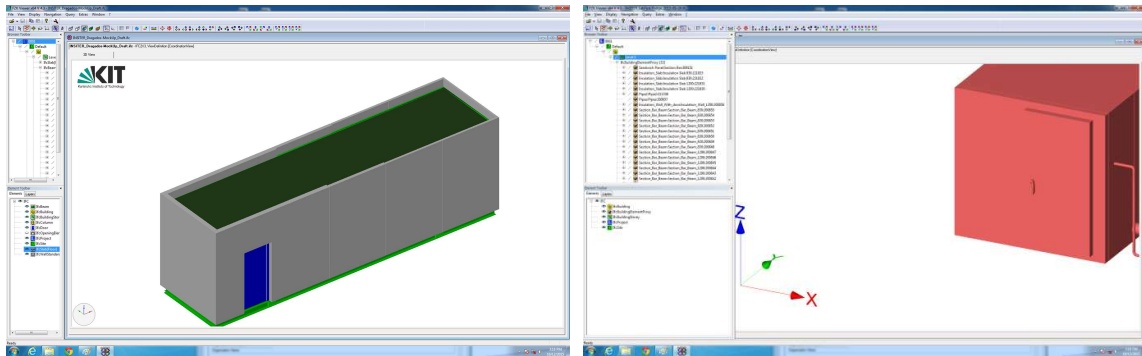


Figure 9, screenshots of FZK Viewer.

Other applications from the same research institute:

- IfcCheckingTool, FZK is the specialist in testing and checking IFC files, this is a tool to help understanding where potential issues in IFC files are located.
- Similar to eveBIM from CSTB are the developers of the FZK viewer a research institute

4.2.9 Tekla BIMsight

Download location: <http://www.teklabimsight.com/>.

This is a free viewer offered by Tekla. Tekla is developer of a CAD package with the focus on steel construction. Tekla is owned by Trimble, one of the largest software companies for Building & Construction. Typical Functionality:

- Clipping and Clash detection
- Support for other formats

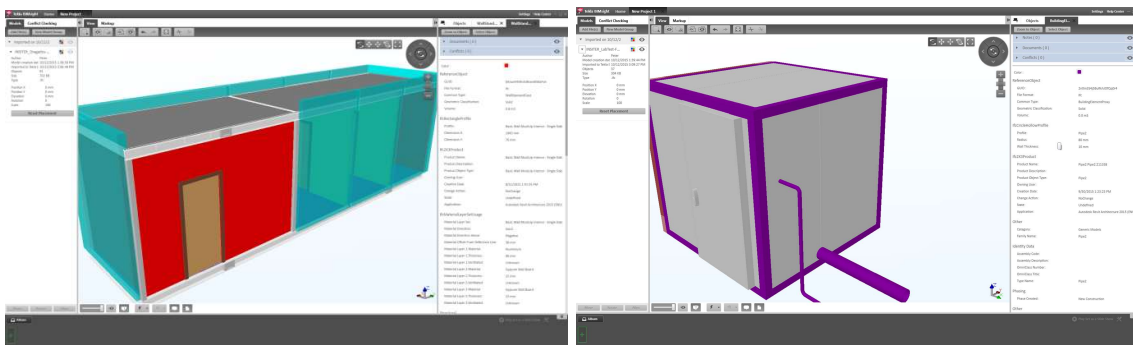
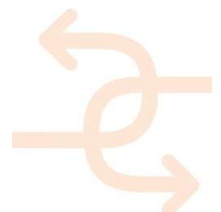


Figure 10, screenshots of Tekla BIMsight.

Other applications from the same vendor (mother company Trimble has much more applications like VICO, SketchUp):

- Tekla, a CAD package mainly focused on steal constructions



4.2.10 **RDF IFC Viewer**

Download location: <http://www.ifcviewer.com/>.

This is a free viewer offered by IFC Engine DLL Toolbox developer. Typical functionality:

- Small and fast
- All source code available

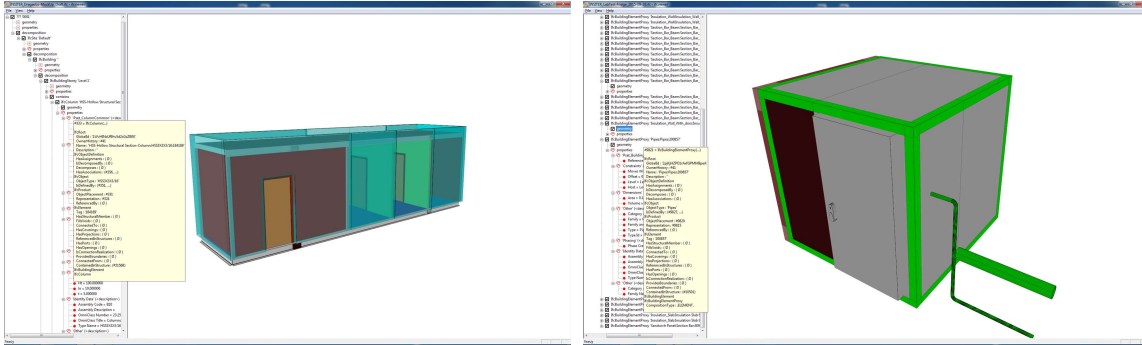


Figure 11, screenshots of RDF Viewer.

Other applications from the same vendor:

- IFC Engine DLL, an IFC Toolbox, multiplatform
- Parametric Real-Time Geometry Kernel, a Geometry Kernel focused on parametric modelling

4.2.11 **Solibri Model Viewer**

Download location: <http://www.solibri.com/products/solibri-model-viewer/>.

Solibri is a software company specialized in modelchecking of IFC files. The Solibri Model Checker is the reference on model checking in the BIM world at this moment and the Solibri viewer is by many seen as the reference viewer of IFC files. Like all its competitors it does not show all IFC files correctly and has issues loading some of them but still can beat (most of its) the competition in correctness. Typical functionality:

- JAVA based
- Often used as reference implementation
- Properties shown complete including derived properties

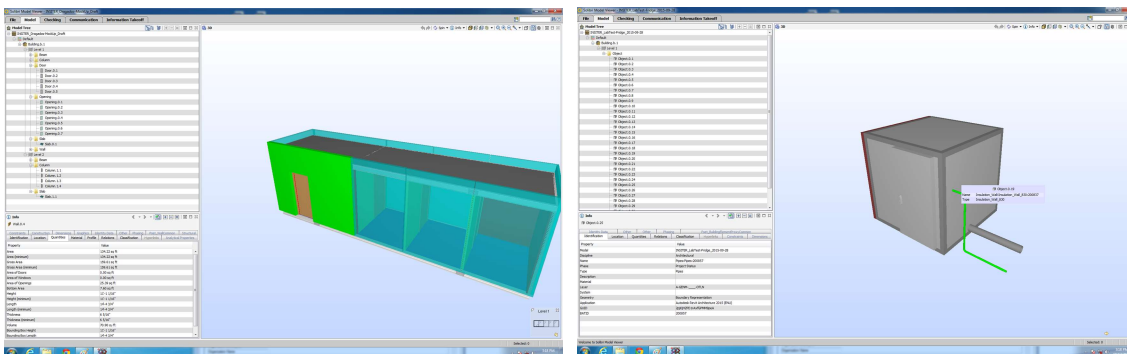


Figure 12, screenshots of Solibri Model Viewer.

Other applications from the same vendor:

- Solibri Model Checker, the reference application for applying automated checks on IFC models.



4.2.12 **Xbim Xplorer**

Download location: <http://www.openbim.org/>.

XBIM is an open source toolbox based on C#, it uses parts of other open source projects. Typical Functionality:

- Source code available

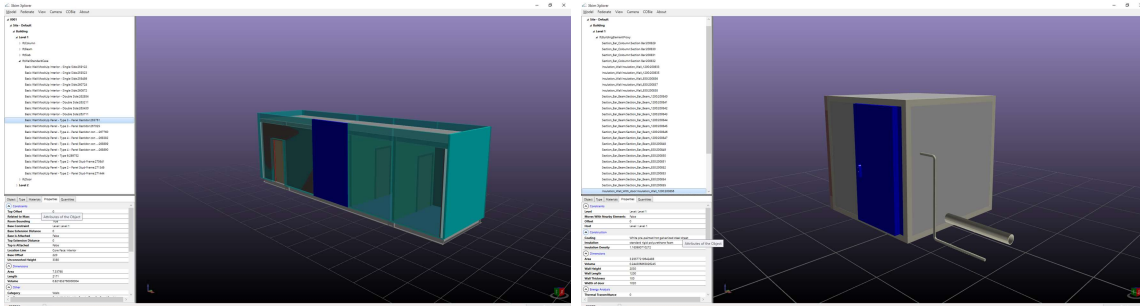


Figure 13, screenshots of Xbim Xplorer.

Other applications from the same vendor:

- The xBIM Toolkit, an open source toolkit written in C# for IFC.

Note: we did not include online viewers like the IFC TOOLS Project viewer or the WebSurfer solution. We found 4 online viewers free to use (both openbim and RDF also have a WebGL based viewer available). The IFC TOOLS Project also has a stand-alone viewer, the viewer application is requested however too late to be included in above list.

Almost all viewers are able to visualize the tree structure from IFC containing both 'decomposition' and 'contain' relationships on the objects within an IFC file. Also almost all viewers are able to visualize properties of each individual object. The way of visualizing object tree and properties differs per viewer.

Functionality we find in some of the viewers is defining clipping planes and support for clash detection. Also other file formats can sometimes be supported. Found issues on our small test cases area:

- Incorrect geometry and placement
- Differences in colors

In conclusion there are several choices possible, the best choice what viewer to use for the framework depends on needed functionality, needed API and programming language, is an open source development relevant and is a desktop version preferred over an Web View solution.

4.3 BIM Server Applications

Next to stand alone viewers there are also BIM server applications. Where BIM (or better said IFC) viewers are normally free to download and use BIM server solutions are normally not free to use. One exception here is the Open Source BIM server mainly developed by TNO and TU/e.

We looked for solutions that support or strongly integrate IFC models within their system. There are three different groups of BIM server solutions divided based on their origin. The BIM server solutions are either coming from existing database solutions applied and extended, they are created by a CAD vendor that sees an growing role for server based



solutions next to their existing CAD systems or they are created by companies active in supporting project management of large and/or complex projects and see a growing need and availability of BIM related data.

The following BIM server solutions originated from CAD:

- BIM+
- BIMCloud
- BIM 360 Glue

The following BIM server solutions originated from Databases:

- Open Source BIM server
- EDM Model Server

The following BIM server solutions originated from Project Management:

- Adoddle
- BIMS SYNC
- Conject
- Thinkproject

Within INSITER currently the solutions of BIM+ and BIMS SYNC are being tested and compared. Both systems are strongly focused on IFC and support applications from different vendors. For both solutions we have access to a test version where we can upload our own models and access the API calls. The Open Source BIM server looks in many ways similar to both BIM+ and BIMS SYNC and is known by several of the partners already and could become a third option to look at.

4.3.1 BIM+

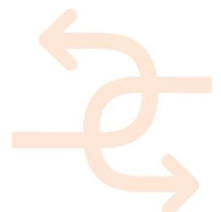
Download location: <http://www.bimplus.net/>.

BIM+ is owned and developed by Allplan GmbH, itself being part of Nemetschek. Nemetschek is one of the largest software developers in the Building & Construction industry and covers both Allplan and ArchiCAD among many other software products. The other three large software companies in the building & construction industry are Autodesk, Trimble and Bentley.

4.3.2 BIMCloud

Download location: <http://www.graphisoft.com/bimcloud/>.

The BIMCloud solution is a Graphisoft solution; Graphisoft is a part of Nemetschek like Allplan/BIM+. BIMCloud is in some sense the modern improved version of the BIM Server solution Graphisoft is deploying since 2009. Although BIMCloud (as well as BIM Server) is supporting IFC as exchange format the solution is strongly focused and featured towards ArchiCAD and BIMx (both products from Graphisoft). A strong integration with dedicated applications has clear benefits for the users of these applications, i.e. working together on the same model, setting restrictions and many other typical server functionality just works more natural from within the application. In this sense BIM+ looks more open and vendor independent in first sight.



4.3.3 BIM 360 Glue

Download location: <http://www.autodesk.com/products/bim-360-glue/>.

The solution of Autodesk for cloud-based BIM management and collaboration. Autodesk is one of the major software vendors for the Building & Construction industry, next to Nemetschek and Trimble.

Similar to BIMCloud this Autodesk BIM 360 Glue is a cloud based solution. One of the main differences with most of the other BIM server applications is the large amount of file types this solution supports and understands.

4.3.4 Open Source BIM server

Download location: <http://www.bimserver.org/>.

Developed mainly by TNO and TU/e it is based on other open source components to store data and arrange import/export of IFC files. The main differences to the other solutions is that this is an open source solution without a commercial company behind it enabling dedicated support and it is somehow more focused on the typical server based functionality and less on individual consumers wishes.

4.3.5 EDM Model Server

Download location: <http://www.epmtech.jotne.com/products/edm-model-server-ifc>.

The EDM Model Server was most probably the first BIM model server created. It is created by EPM Jotne, a software company focused on technical high-end solutions for supporting STEP standards. The company is active in the automotive and aerospace industry and creates state-of-the-art solution for storing large amounts of technical data.

Several well-known CAD packages use the toolkit from the same company to create and parse IFC data.

4.3.6 Adoddle

Download location: <https://www.asite.com/adoddle/corporate-collaboration>.

Adoddle is the main solution of ASite. It is a collaboration environment with support for BIM via supporting IFC serialization.

4.3.7 BIMSYNC

Download location: <http://www.bimsync.com/>.

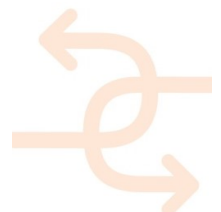
BIMSYNC is developed and supported by Catenda (www.catenda.no). Catenda is a spin-off of SINTEF, the largest independent research organization in Scandinavia. Catenda is a strong driver behind bsDD (Building Smart Data Dictionary); bsDD is part of the BuildingSMART organization next to IFC. BIMSYNC has an API for accessing geometrical elements and an API to access the database functionality itself.

4.3.8 ILM

Download location: <https://www.conject.com/en/at/home>.

Infrastructure Lifecycle Management (ILM) is a product to support the lifecycle of large and complex projects; the product is the main software product of Conject (<http://www.conject.com>). Recently Conject added BIM support to their product.

From open BIM perspective it is a very welcome development that BIM is now integrated and supported in such products, the use of such a specific product within INSITER seems to be less relevant as BIM support is really focused on the running projects and ILM flavored.



4.3.9 Think Project!

Download location: <https://www.thinkproject.com/en/produkt/bim-collaboration/>.

Think project! is a cloud solution for cross-enterprise collaboration specifically tailored towards the requirements of construction and engineering projects. The company behind carries the same name as its main product (<http://www.thinkproject.com>). ThinkProject is developing and integrating BIM support for many years already and is now actively using it in their projects. This solution is dedicated to ThinkProject and really focused on running projects.

As a conclusion it seems the most interesting solutions for INSITER are the ones with an open API, focus on vendor independent solutions and extendibility. Currently the following BIM servers application are checked in more detail:

- BIM+
- BIMSYNC
- Open Source BIM server

Access to the API's was granted for INSITER by the companies behind. The functionalities of each solution is looked at in more detail. Based on the requirement and use-cases the best solution will be chosen. It is expected none of these servers can fulfill all the requirements for storing BIM data. As most probably data will be available in other formats not supported by the selected server a file server solution like for example SharePoint will be expected to be needed next to the BIM server.

4.4 Self-Instruction and Self-Inspection Applications

It is the goal to create Self-Instruction and Self-Inspection applications within INSITER.

Self-instruction and self-inspection applications are expected to make use of the BIM models available and interact with the user with help of Augmented Reality. The self-instruction and self-inspection applications are expected to become part of a project independent Instruction model layer making use of documents available.

The self-instruction and self-inspection models are therefore project independent i.e. can be used for different projects. Within INSITER we will focus on how open standards like IFC can be used to support such self-instruction and self-inspection applications.

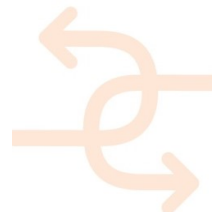
4.5 BIM Services

Services are a subset of applications that are able to run stand-alone without user interaction. A typical service could be doing an Energy-Calculation. In the context of the hardware devices this could be processing of the raw data.

Services can be triggered by applications that use a service to generate information they want to show to the user but alternatively services can also be triggered by the data store when a certain information source is updated.

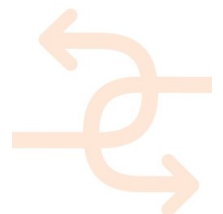
We identified the following services:

- Energy Calculation and Energy Simulation
- Clash Detection



- Model Checking
- 4D Issue Tracking
- Constructability
- Logistics
- Processing data from hardware.

Part of these services could be applications also. However in many cases the service itself can be called by the accompanying application to clearly separate the processing and user interaction.



5. Architecture - Framework

5.1 Framework - Components

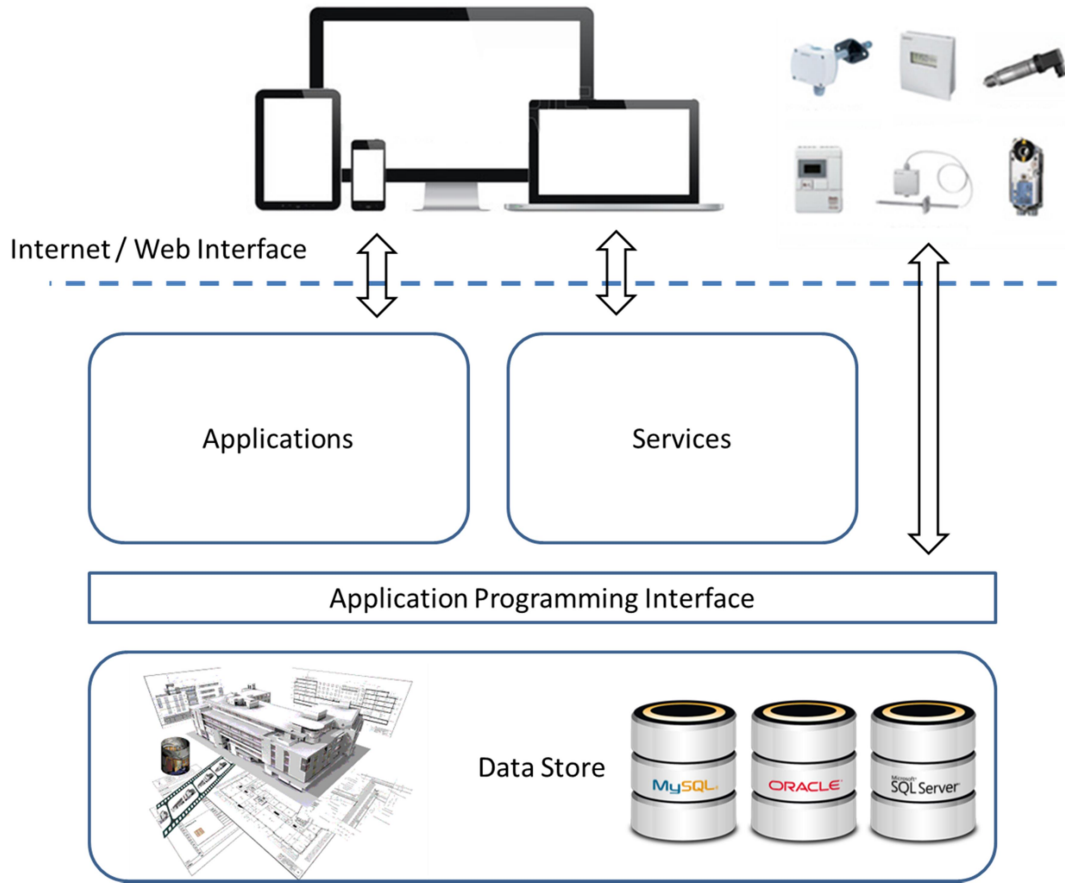


Figure 14, Framework.

5.1.1 Client Side – PC / Laptop / Tablet / Phone

All applications are actually executed server side, this means the client side works as a 'thin client' via a web browser.

5.1.2 Client Side – measurement devices

Measurement devices are located at client side, the actual device is clearly used outside the server, however services or applications to interpret the raw data and convert them to sensible images or even a BIM model can very well be part of the applications and services on server side, however they could also run on a dedicated client machine. Although it is expected that raw data information or processed information are stored on the server, it is possible that client devices like tablets or phones could trigger the measurement devices to directly start and scan information . For example a scan can be shown directly on a tablet integrated with an Augmented Reality application.



5.1.3 Server Side – applications

Stand-alone applications will be running server side. This means somehow the interface of the applications has to be exposed to the client via a web browser. To establish such architecture the framework uses a Terminal Server Solution. Web enabled applications do not need this Terminal Server Solution and can arrange their own interface to the client via a web browser.

5.1.4 Server Side – services

Services are small applications running on the server without an interface to the client. Some services can be triggered automatically by updates on the server, i.e. the server will decide when a service has to run. In other cases the client or an application can trigger the service to execute. Results of the server can either be fed back to the data store via the API or shown directly via an application interacting with the client and visualizing the results generated by the service.

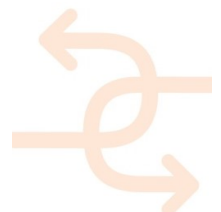
5.1.5 Server Side - API

An application programming interface (API) is enabling access from applications and services towards the data store. Also measurement devices can directly access the API to store scanned information in the data store directly. The structure of an API and calls that can be found are normally depending on the technical solution selected, i.e. every software solution chosen has its own API and therefore its own limitations. Although there are some exceptions, for example SDAI is an ISO standardized vendor independent API, normally the choice of software product for the data-store decides the API available and limitations/freedom in which applications and services can arrange a connection to the data store.

5.1.6 Server Side - Data Store

As we already found in the different BIM server solutions a server can be either a single server machine or a cloud based solution. The technical solution how data is distributed (a cloud server or single server) does not influence the API functionality. Within data stores there is a difference between natively supported formats and generic raw data files. In case of generic raw data files this could be any file from picture, movie and text file towards a BIM model or even the application installer itself. Servers with a focus on file storage rather than natively supported formats have functionality for versions, owner and accessibility rights sometimes backed up by meta-information on the files itself. In case certain file formats are supported natively the server understands and recognizes the individual objects and instances within the file itself, this enables updating and right management on parts of the files. In chapter 4, the part about BIM servers, we are looking at data store solutions (including an API) that understand the file format IFC.

For point cloud data there exists also dedicated cloud solutions and spatial data bases. The data store solution could exist of a set of data bases.



5.2 Framework – BIM

The Framework components will be further detailed with selected components. The data-store component will most probably get a BIM server as part of its architecture. The following picture shows a more detailed architecture of the data store with a BIM server in place.

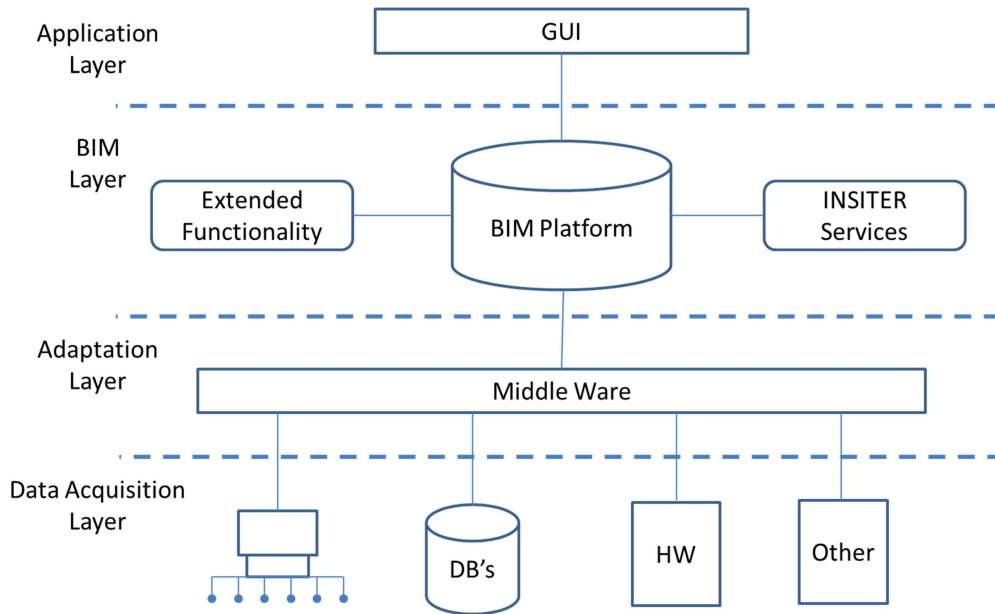


Figure 15, Framework - BIM Server.

5.2.1 GUI

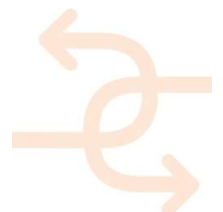
The application layer contains the applications, services and API from the Framework Components. The Graphical User Interface is the actual interface to the user of the framework.

5.2.2 BIM Platform

The most important part of the BIM layer is the BIM platform. The exact platform is not yet selected at this stage, paragraph 4.3 describes in more detail the different options in the market and 3 of these solutions are investigated in more detail. The actual selection of BIM Platform for INSITER depends on requirements and use-cases. It is expected the BIM server will be accompanied with a file server.

5.2.3 Extended Functionality

As it is expected that no solution in the market will provide all functionality required from a BIM platform extended functionality is needed. Extended functionality can be created on top of the API delivered by the BIM platform, several of the available platforms are prepared for user specific third party extensions.



5.2.4 INSITER Services

Where extended functionality is adding typical server functionality to the BIM Platform, INSITER services make use of the server (extended) functionality to apply and enable typical INSITER related services on top of the BIM data.

5.2.5 Middle Ware

The Middle adaptation layer will add will connect all kind of information sources to the BIM platform. The data acquisition layer (seen from BIM platform point of view) will be able to get relevant information from other existing databases, hardware sensors and other sources like existing libraries or processed data from sensors.

5.3 Framework - File Types and Formats

The Framework solution will need to understand a large variety of different file types. There are loads of different file types such as:

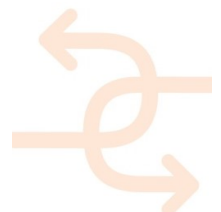
- Point Clouds
- Images
- Movies
- BIM models
- Other Data files

Each file type has many file formats. For example images can be stored as GIF, BMP, JPEG, etc. BIM models can be stored as .rvt, gbXML, IFC etc.

Each file format has several releases, for example GIF (Graphics Interchange Format) has a GIF 87 specification, however also a GIF 89a specification with Graphics Control Extension (GCE). The same goes for IFC, IFC 1.5.1, IFC 20 LF, IFC 2x, IFC2x2, IFC 2x3, IFC 2x3 TC1, IFC4, etc.

Within each release applications often support only a part of the data. Within IFC there is a technique to define what part of the IFC schema is supported by a software. It is called a Model View Definition (MVD), as each discipline has specific knowledge it wants to exchange the subset of the IFC schema differs per discipline.

Chapter 2 describes the difference between open standards and closed BIM, within INSITER we will focus on open standards as much as possible. For BIM CAD models this means focusing on file format IFC, the most relevant releases are IFC 2x3 TC1 and IFC4, IFC4 TC1 will probably become relevant also during the scope of INSITER. For files coming from the hardware several other formats can be expected.



5.4 Connecting Different Data

In many cases different data sources stored in different file formats will need to be connected somehow. As within the INSITER Framework the geometric BIM model plays an important role as central model within the data-store. This paragraph looks at connecting other data sources towards an IFC model.

There are 4 different architectural solutions:

- IFC referencing to external data sources
- External data sources referencing to IFC
- Independent data sources defining interlinks between data sources + IFC ('Linked Data' technology solution)
- External data embedded within IFC

5.4.1 IFC referencing to external data sources

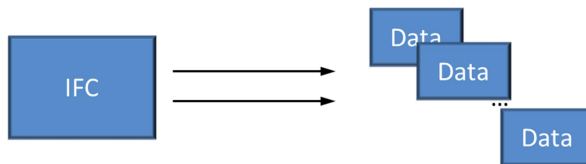


Figure 16, IFC => Data.

IFC contains placeholders for references to external data sources/libraries.

The benefit of this solution is that some applications could understand if external sources exist and are referenced from within IFC. See also appendix 1. The draw-back of this solution is that the IFC file has to be adjusted, as the linking information need to be added within the IFC file, i.e. sync issues when CAD generates a revision of the IFC content.

5.4.2 External data sources referencing to IFC

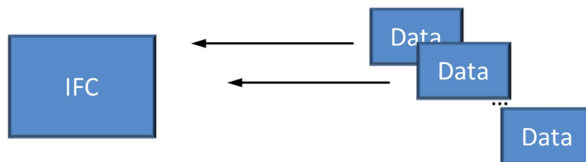


Figure 17, IFC <= Data.

External data sources/libraries contain the references to IFC (and potentially between themselves).

This seems to be a natural solution as we create/access/convert the data sources anyway. The drawback is that it has to adjust data sources, i.e. sync issues when a revision of the data source is given.



5.4.3 Independent data sources defining interlinks between data sources + IFC ('Linked Data')

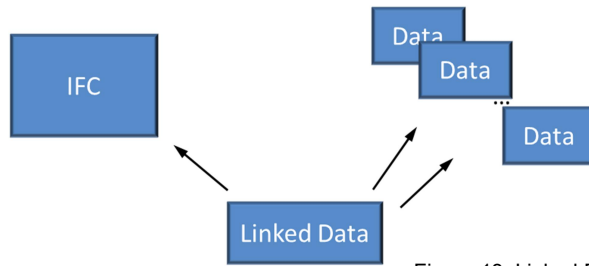


Figure 18, Linked Data.

An independent data source contains the relations between IFC and data sources. The original files are unchanged. This is the preferred solution for INSITER.

The benefit of this solution is that it does not generate (limited) conversions (i.e. normally explained as no/less data loss); less sync issues. The drawback is that it generates more different formats to be supported by software, complexity is moved to software solutions.

Note I: this type of architecture is often called 'Linked Data Technology'.

Note II: normally Linked Data contains just references, we discussed not limiting us to that at this moment. The linked data file could contain also geometry or even be an IFC file. ('Linked Data Technology' solutions will probably have specific ideas about this and prefer to separate data and metadata)

5.4.4 External data embedded within IFC

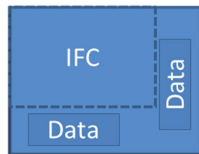


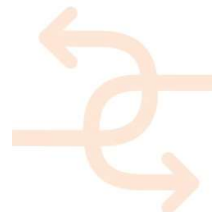
Figure 19, Embedded Data.

Data is embedded within IFC. Where possible IFC entities and Pset property sets will be used to store data. In some cases this can be a preferred solution.

The benefit of this solution is that existing software solutions will better 'understand' the content/data; one clear file as a result. The draw-back is data/knowledge loss and complicated sync issues.

Two examples of this last option:

- In the appendix there is a description how to integrate knowledge from sensors in an IFC model, if possible such knowledge will/could be understood by the application and therefore has benefit over dedicated INSITER solutions.
- Also the research from other projects is very relevant, the Duraark project did detailed developments in the area of integrating point clouds within IFC files. The use of such knowledge is depending on the typical use cases we will find within INSITER as well as the hardware that will be selected.



6. Conclusions and Outlook

6.1 Conclusions

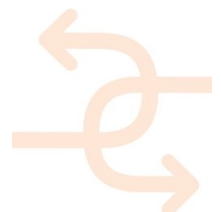
The abstract Framework is now available for use. The drawback of requesting this framework early in the project is that it is complicated to finalize a solution based on limited requirements that will have to fit all use-cases that have to be developed still. The benefit is that the result can be used for all developments within the project.

During review and tests of different software packages based on the IFC models already created in task 4.1, it was found that several applications can be found for both IFC viewers and IFC / BIM server solutions. However, the quality and functionality of the solutions have important differences, even on relatively small and simple models, several mistakes in applications were identified. A good review before selection is required.

6.2 Outlook

The current testing of BIM servers BIM+, BIMSIGHT and the Open Source BIM server is expected to lead to the selection of the application used as data-store or more precise as BIM Platform.

Once the use-cases are selected and hardware sensor devices are selected a more detailed set of requirements is expected to arise. This will be used to review the current abstract Framework and will result in selection of applications for the different components in the Framework. Also data-flow and data-formats used can be more detailed.



APPENDIX 1 – Sensor Data in IFC

The new standard IFC4 contains the entity `IfcSensor` and its type `IfcSensorType`. See also pictures below.

`IfcSensor` seems to be meant for installed sensors. Most of the sensors that need to be supported for self-inspection within INSITER are temporary sensors. The question is how to look at the sensors that are used temporary:

- Even temporary sensors can be modelled by `IfcSensor` as they are actually inside the building in a certain stage. The IFC model is used as a representation of the building at time X where the measurement is done.
- Temporary sensors are not part of the IFC model, however we can connect the data they produce. This data can be connected as an external data set to `IfcSpace` or `IfcZone`.
- Hybrid solution.

External Libraries

External libraries can be connected in IFC2x3 already; however possibilities (especially for external datasets) are extended in IFC4.

Especially of interest are:

`IfcLibraryReference`

`IfcDocumentReference` (`IfcDocumentInformation`)

This functionality can be used to link/reference datasets created by the sensor devices.

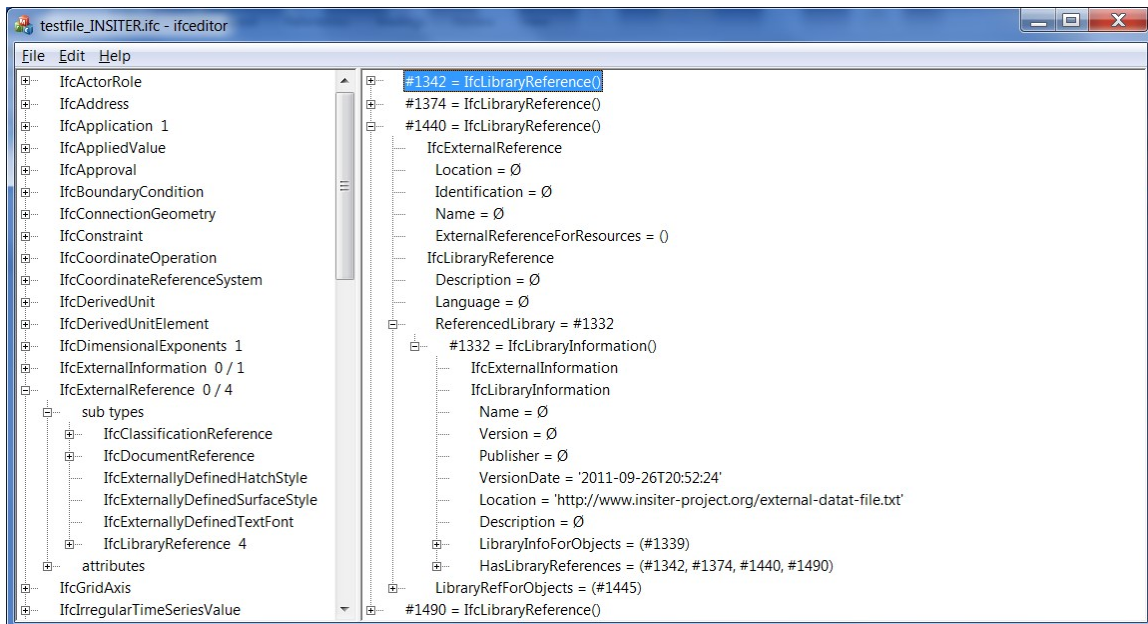
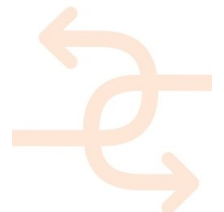


Figure 19, `IfcLibraryReference`.



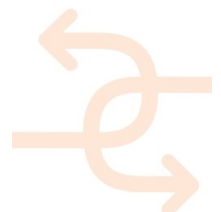
IfcSensor

This is the entity in IFC4 that defines the instances. Instances can contain geometry, location, relations like Walls, Windows, Slabs, etc. This information is IFC4 schema defined. IfcSensor defines the occurrence of any sensor; common information about sensor types is handled by IfcSensorType. The IfcSensorType (if present) may establish the common type name, usage (predefined type), properties, materials, ports, composition, assignments, and representations. The IfcSensorType is attached using the IfcRelDefinesByType objectified relationship and is accessible by the IsTypedBy inverse attribute. If the IfcSensorType has ports or aggregated elements, such objects are reflected at the IfcSensor occurrence using the IfcRelDefinesByObject relationship.

```

ENTITY IfcSensor;
ENTITY IfcRoot;
  GlobalId      : IfcGloballyUniqueId;
  OwnerHistory  : OPTIONAL IfcOwnerHistory;
  Name          : OPTIONAL IfcLabel;
  Description   : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
  HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
  Nests          : SET [0:1] OF IfcRelNests FOR RelatedObjects;
  IsNestedBy    : SET OF IfcRelNests FOR RelatingObject;
  HasContext    : SET [0:1] OF IfcRelAssignsToContext FOR RelatedDefinitions;
  IsDecomposedBy : SET OF IfcRelAggregates FOR RelatingObject;
  Decomposes    : SET [0:1] OF IfcRelAggregates FOR RelatedObjects;
  HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
  ObjectType    : OPTIONAL IfcLabel;
INVERSE
  IsDeclaredBy  : SET [0:1] OF IfcRelDefinesByObject FOR RelatedObjects;
  Declares      : SET OF IfcRelDefinesByObject FOR RelatingObject;
  IsTypedBy     : SET [0:1] OF IfcRelDefinesByType FOR RelatedObjects;
  IsDefinedBy   : SET OF IfcRelDefinesByProperties FOR RelatedObjects;
ENTITY IfcProduct;
  ObjectPlacement : OPTIONAL IfcObjectPlacement;
  Representation   : OPTIONAL IfcProductRepresentation;
INVERSE
  ReferencedBy  : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
  Tag           : OPTIONAL IfcIdentifier;
INVERSE
  FillsVoids    : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
  ConnectedTo   : SET OF IfcRelConnectsElements FOR RelatingElement;
  IsInterferedByElements : SET OF IfcRelInterferesElements FOR RelatedElement;
  InterferesElements : SET OF IfcRelInterferesElements FOR RelatingElement;
  HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
  ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
  HasOpenings   : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
  IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR
    RealizingElements;
  ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
  ConnectedFrom   : SET OF IfcRelConnectsElements FOR RelatedElement;
  ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR
    RelatedElements;
ENTITY IfcDistributionElement;
INVERSE
  HasPorts      : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
ENTITY IfcDistributionControlElement;
INVERSE
  AssignedToFlowElement : SET [0:1] OF IfcRelFlowControlElements FOR
    RelatedControlElements;
ENTITY IfcSensor;
  PredefinedType : OPTIONAL IfcSensorTypeEnum;
END_ENTITY;

```



IfcSensorTypeEnum

This is the enumeration of types of sensors. This is an alternative for subclassing further the entity IfcSensor. These values are IFC4 schema defined.

CONDUCTANCESENSO - A device that senses or detects electrical conductance.

CONTACTSENSOR - A device that senses or detects contact, such as for detecting if a door is closed.

FIRESENSOR - A device that senses or detects fire.

FLOWSENSOR - A device that senses or detects flow in a fluid.

GASENSOR - A device that senses or detects gas concentration.

HEATSENSOR - A device that senses or detects heat.

IONCONCENTRATIONSENSOR - A device that senses or detects ion concentration, such as for water hardness.

LEVELSENSOR - A device that senses or detects fill level, such as for a tank.

HUMIDITYSENSOR - A device that senses or detects humidity.

LIGHTSENSOR - A device that senses or detects light.

MOISTURESENSOR - A device that senses or detects moisture.

MOVEMENTSENSOR - A device that senses or detects movement.

PHSENSOR - A device that senses or detects acidity.

PRESSURESENSOR - A device that senses or detects pressure.

RADIATIONSENSOR - A device that senses or detects electromagnetic radiation.

RADIOACTIVITYSENSOR - A device that senses or detects atomic decay.

SMOKESENSOR - A device that senses or detects smoke.

SOUNDSENSOR - A device that senses or detects sound.

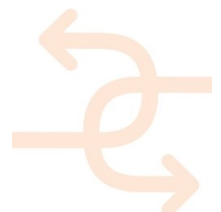
TEMPERATURESENSOR - A device that senses or detects temperature.

WINDSENSOR - A device that senses or detects airflow speed and direction.

USERDEFINED

NOTDEFINED

Every element above except for MOISTURESENSOR, USERDEFINED and NOTDEFINED has a dedicated property set with properties belonging typically to sensors of such type. The property sets are not defined in the IFC4 schema, however they are part of the IFC4 definition.



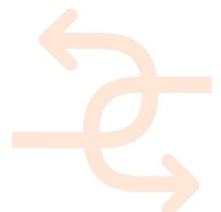
lfcSensorType

This is the type of sensor. When several sensors share properties, behavior etc. this data can be attached to the lfcSensorType.

```

ENTITY IfcSensorType;
  ENTITY IfcRoot;
    GlobalId      : IfcGloballyUniqueId;
    OwnerHistory  : OPTIONAL IfcOwnerHistory;
    Name          : OPTIONAL IfcLabel;
    Description    : OPTIONAL IfcText;
  ENTITY IfcObjectDefinition;
  INVERSE
    HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
    Nests          : SET [0:1] OF IfcRelNests FOR RelatedObjects;
    IsNestedBy    : SET OF IfcRelNests FOR RelatingObject;
    HasContext     : SET [0:1] OF IfcRelAssignsToContext FOR RelatedDefinitions;
    IsDecomposedBy : SET OF IfcRelAggregates FOR RelatingObject;
    Decomposes    : SET [0:1] OF IfcRelAggregates FOR RelatedObjects;
    HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
  ENTITY IfcTypeObject;
    ApplicableOccurrence : OPTIONAL IfcIdentifier;
    HasPropertySets      : OPTIONAL SET [1:?] OF IfcPropertySetDefinition;
  INVERSE
    Types : SET [0:1] OF IfcRelDefinesByType FOR RelatingType;
  ENTITY IfcTypeProduct;
    RepresentationMaps : OPTIONAL LIST [1:?] OF UNIQUE IfcRepresentationMap;
    Tag                : OPTIONAL IfcLabel;
  INVERSE
    ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
  ENTITY IfcElementType;
    ElementType : OPTIONAL IfcLabel;
  ENTITY IfcDistributionElementType;
  INVERSE
    HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
  ENTITY IfcDistributionControlElementType;
  ENTITY IfcSensorType;
    PredefinedType : IfcSensorTypeEnum;
END ENTITY;

```

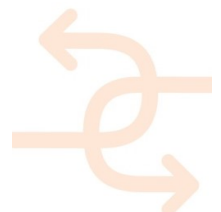


IfcLibraryReference

A reference to an external library.

```
ENTITY IfcLibraryReference
  ENTITY IfcExternalReference
    Location :OPTIONAL IfcURIReference;
    Identification :OPTIONAL IfcIdentifier;
    Name :OPTIONAL IfcLabel;
  INVERSE
    ExternalReferenceForResources: SET [0:?] OF IfcExternalReferenceRelationship FOR
      RelatingReference;

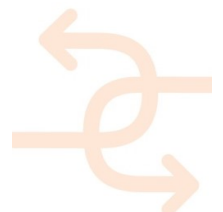
  ENTITY IfcLibraryReference
    Description :OPTIONAL IfcText;
    Language :OPTIONAL IfcLanguageId;
    ReferencedLibrary :OPTIONAL IfcLibraryInformation;
  INVERSE
    LibraryRefForObjects: SET [0:?] OF IfcRelAssociatesLibrary FOR RelatingLibrary;
END_ENTITY;
```



lfcDocumentReference

A reference to an external document.

```
ENTITY IfcDocumentReference
ENTITY IfcExternalReference
  Location :OPTIONAL IfcURIReference;
  Identification :OPTIONAL IfcIdentifier;
  Name :OPTIONAL IfcLabel;
INVERSE
  ExternalReferenceForResources: SET [0:?] OF IfcExternalReferenceRelationship FOR
    RelatingReference;
ENTITY IfcDocumentReference
  Description :OPTIONAL IfcText;
  ReferencedDocument :OPTIONAL IfcDocumentInformation;
INVERSE
  DocumentRefForObjects: SET OF IfcRelAssociatesDocument FOR RelatingDocument;
END_ENTITY;
```



IfcDocumentInformation

Information about a referenced document as available in IFC4.

```

ENTITY IfcDocumentInformation
  ENTITY IfcExternalInformation
  ENTITY IfcDocumentInformation
    Identification :IfcIdentifier;
    Name           :IfcLabel;
    Description    :OPTIONAL IfcText;
    Location       :OPTIONAL IfcURIReference;
    Purpose        :OPTIONAL IfcText;
    IntendedUse    :OPTIONAL IfcText;
    Scope          :OPTIONAL IfcText;
    Revision       :OPTIONAL IfcLabel;
    DocumentOwner :OPTIONAL IfcActorSelect;
    Editors        :OPTIONAL SET [1:?] OF IfcActorSelect;
    CreationTime   :OPTIONAL IfcDateTime;
    LastRevisionTime :OPTIONAL IfcDateTime;
    ElectronicFormat :OPTIONAL IfcIdentifier;
    ValidFrom      :OPTIONAL IfcDate;
    ValidUntil     :OPTIONAL IfcDate;
    Confidentiality :OPTIONAL IfcDocumentConfidentialityEnum;
    Status         :OPTIONAL IfcDocumentStatusEnum;
  INVERSE
    DocumentInfoForObjects:SET OF IfcRelAssociatesDocument FOR RelatingDocument;
    HasDocumentReferences :SET OF IfcDocumentReference FOR ReferencedDocument;
    IsPointedTo           :SET OF IfcDocumentInformationRelationship FOR RelatedDocuments;
    IsPointer              :SET [0:1] OF IfcDocumentInformationRelationship FOR RelatingDocument;
END_ENTITY;

```

