INSITER INTUITIVE SELF-INSPECTION TECHNIQUES

Functional Programme of Requirements for building quality and energy assessment software

Deliverable report 3.3

Deliverable Report: FINAL 07 October 2016. Revised and resubmitted to address EC comments.

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.

INSITER
INTUITIVE
SELF-INSPECTION
TECHNIQUES

Functional Programme of Requirements for building quality and energy assessment software

Deliverable report 3.3

Issue Date	7 October 2016
Produced by	Work Package 3 Team (WP leader: DMO)
Main authors	Richard Deighton (DMO), Gaby Abdalla (DWA)
Co-authors	André van Delft (DMO), Peter Bonsma (RDF)
Version:	Final. Revised and resubmitted to address EC comments.
Reviewed by	Klaus Luig (3L), Jan-Derrick Braun (HVC)
Approved by	Rizal Sebastian (Technical Coordinator, DMO)
Dissemination	Public

Colophon

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.

Fulfilment of the Description of Action (DoA) by D3.3

Summarised objectives as stated in DoA	Results in this deliverable addressing the DoA
WP 3 scope and objectives: - To develop software applications for self-inspection based on the advancements of existing software tools for quality and energy-performance assessment, planning and cost monitoring, and decision-support systems. - BIM and self-inspection and self-instruction interfaces in software tools.	Addressed: This deliverable presents the software development requirements, software architecture and proof-of-concept including examples of the intended features. The INSITER software will be developed based on state-of-the-art IT tools such as SharePoint, RE Dashboard, RE Maintenance, Desite MD and Navisworks. These tools are analysed and described in this deliverable. This deliverable thus fulfils 50% of the WP3 scope and objective whereas the remaining 50% will be fulfilled in the follow-up deliverable D3.4. Not addressed : Planning and cost monitoring is not within the scope of this deliverable, but described in the complementary D3.1.
Task 3.2 scope and objectives: - Reviewing the state-of-the-art software tools for building condition assessment and the tools for energy and comfort assessment. - Bridging the gap between the assessment toolset to diagnose building condition and the assessment toolset to measure energy performance and comfort. - Developing interfaces between the software tools and BIM, and the Building Maintenance Systems (BMS). - Embedding Key Performance Indicators, calculation, and measurement and verification protocols in the software tools; defining algorithms for taking data and evaluating them against the set of Key Performance Indicators. - Analysing a set of scenarios and recommending the most optimal solutions using the decision support systems.	Addressed: • State-of-the-art software tools are critically reviewed in this deliverable (chapter 5). This topic has been achieved 100%. • Proposed solutions to bridging the gap between software for assessment of building condition and energy performance + comfort are presented in this deliverable (chapter 4 and 6). This topic has been achieved 50%; the remaining 50% will be achieved in the follow-up deliverable D3.4 where the prototype software will be demonstrated. • Proposed solution is explain in which the energy performance results will be simulated and imported into the BIM server via tools like Vabi Elements, TRNSYS, Simaxx. Implementation in BMS is part of building operation / monitoring and as such it is beyond the scope of INSITER since INSITER focuses on construction / renovation / maintenance processes. This topic has been achieved 50%; the remaining 50% will be achieved in the follow-up deliverable D3.4 where the prototype software will be demonstrated. • The way for embedding Key Performance Indicators, calculation, and measurement and verification protocols, as well as decision-support mechanism with scenario analysis, in the software tools is described in this deliverable (chapter 4 and 6). This topic has been achieved 50%; the remaining 50% will be achieved in the follow-up deliverable D3.4 where the prototype software will be demonstrated.
Deliverable D3.3 scope and objectives: Functional requirements and integration architecture of applications for building quality and energy assessments, which will serve as the underpinning for further software development addressing building quality and energy performance.	Specific results fulfilling the deliverable objectives: All aspects are addressed comprehensively in this deliverable, i.e. functional requirements are described in chapter 2, and integration architecture is described in chapter 6. This topic has been achieved 100%.

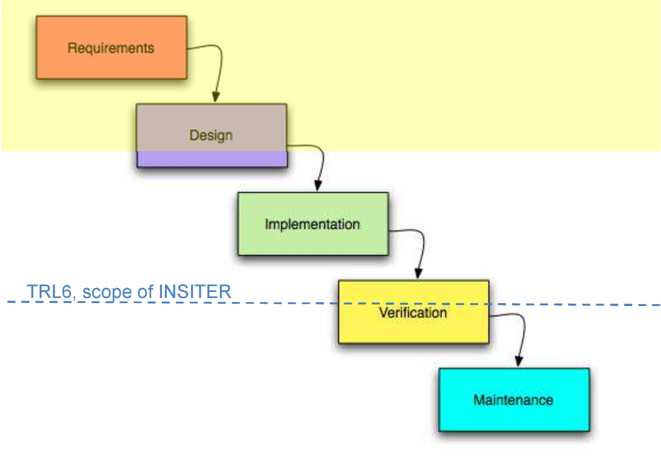
Project's progress relevant to the deliverable within the corresponding timeframe (Year 1 - 2 until M18): - Most relevant software is reviewed to define the functional needs and required interfacing (e.g. DEMO RE Suite for condition assessment and asset management, DWA Simaxx for energy and comfort assessment, tools for scheduling and cost management). - KPIs are integrated in the self-inspection protocols/manuals. Simultaneously, the reviewed and optimised energy performance and condition assessment norms, references and calculation methods are embedded into the analytical software.	This deliverable is 100% aligned with the described progress plan: • DEMO RE Suite and DWA Simaxx have been reviewed for further enhancement to achieve INSITER's goals. Note: In the DoA, DWA's software for MEP system is called "Monavisa". This software has recently become a part of a business entity (a newly established company) named "Simaxx". The collaboration with Simaxx remains a responsibility of DWA. • INSITER KPIs have become an integral aspect of software development. The KPI definition is done in WP1 (beyond the scope of this deliverable); yet embedding KPIs into the software has been dealt with in this deliverable. The configuration of KPIs in prototype software will be part of the follow-up deliverable D3.4.

Publishable executive summary

In an ever digitalizing environment the role of software becomes more and more important. Software is used on PC's and laptops but also many other devices contain software, of course phones and other smart devices but almost any modern device from vacuum cleaner till bread toaster contains software. The development of software therefore is a large and mature business with well-defined processes and development methods.

Concerning software development the first thing many people think about is the software coding itself. Although this is of course an essential part it is not the only part in the process. During the past 30 years and after many wonderful software products able of offering great functionality but totally useless for the client it was developed for it was understood requirements management is at least as important as the actual software coding itself.

The software development will be executed according to the widely acknowledged Waterfall model. This model exists of a set of process steps that are executed in a sequential order as is shown in the scheme below.

Waterfall model	Coverage of current and next deliverables
 Requirements Design Implementation TRL6, scope of INSITER Verification Maintenance	Main coverage of this deliverable D3.3
	Fundamental design covered in this deliverable
	Elaborated design and software prototyping will be the focus of next deliverable D3.4
	Software prototype TRL 6+ will be demonstrated in INSITER; the commercial development and implementation incl. maintenance for the client are part of future exploitation, yet out of scope of R&D in INSITER project

Note that the requirements collection and analysis is part of D3.3 and also a global idea of the functional design is part of D3.3. The actual functional design will be worked out in D3.4.

This deliverable focusses on the requirements for building quality and energy assessments; the document will even go a few steps further by translating these requirements in early functional design documents. We take as much as possible advantage of the already existing software brought in by the consortium partners and enhance it in order to fulfil the needs. This will not only prevent additional costs but it is also easier to make the required enhancements to the software.

The deliverable is created in conjunction with the deliverable D3.1 which handles the cost-estimation and planning aspects. The body of knowledge in this report is structured as follows:

- Following the Introduction chapter, the definition and elaboration of INSITER requirements for software development are described in Chapter 2. The requirements are directly addressing INSITER self-instruction and self-inspection processes as well as practical user aspects of the software.
- In Chapter 3, the requirements are clustered in 2 categories: functional, technical and interoperability requirements. The clustered requirements are further analysed and extended with references.
- Chapter 4 describes the software functionalities of the components relevant for INSITER and developed and/or used in Task 3.2 / D3.4. Over here a details description can be found from the 'Defect and quality issue registration tool' and the new process to use modern stat-of-the-art BIM based energy calculation tools for improving understanding of consequences of deviations found during the inspection process.
- As certain software components can be derived from existing software –either open-source, prototype or commercial packages–, a critical review of state-of-the-art software is presented in Chapter 5. For each functionality, conclusions and decisions have been drawn to set a direction for INSITER software development.
- Software architecture topology and related databases figures are added in chapter 6 to show which functionalities are needed and in what context they will be applied in order to fulfil these requirements. The stakeholders are on one hand the assembly/construction worker that carries out the self-inspections supported with self-instructions, Augmented Reality (AR) and registration facilities for defects and quality issues. On the other hand the project manager monitoring the projects efficiency and risk's via a KPI dashboard.
- The software proof-of-concept is subsequently presented in Chapter 7, and the technical plan for the follow-up development in the upcoming deliverable is presented in Chapter 8.

The analysed and consolidated software requirements can be summarized as follows:

	Functional requirements	Technical requirements	Interoperability requirements
1. Requirements related to self-instruction processes	Workers start their daily work by following project planning and executing assigned activities (as discussed in previous section). Workers benefit from information about construction/assembly activities when to perform (process), how to perform (guidelines and know-how) but also what to perform (product and building specifications).	The toolset should be able: – to show relevant instructions and information related to the assigned activities. This can be done by web-based application which can be used device independent; – to show available documents on a web page and to offer the possibility to download documents, images and movies that are related to a certain part of the BIM model via the Internet.	The toolset should be able to import product specification (database or BIM), building & design specification (BIM) and related guidelines (database). The tools should also be able to import documents and checklists (mostly related to standard processes).

2. **Requirements related to self-inspection processes**	As described in the Section 2.3 workers use the INSITER toolset to perform inspections by themselves. The toolset will act as a connecting platform for the several elements (instruments, sensors) involved in the several stages (do, check and act). In inspection processes, workers use the toolset to register defects and insert/import readings or measurements. They use the toolset to perform model check and clash detections. In some cases, workers use the toolset to export readings to calculation simulation software to decide on the impact of found deviations on KPIs. The export process may vary from automatically (connection though IT solutions) to manually (self-entry) data export. For this reason, we discuss the evaluation and decision making process separately in Section 2.7.	The toolset should be able to: - provide workers with the ability to register defects and quality issues; - provide the assembly/construction with the ability to repetitively verify the construction during realization against the design model; - visualize the design model on-site including HAVC/MEP systems; - provide augmented reality (web-based application) overlaying the reality on site with (parts of) the design model; - provide workers with drawings and other documents; - provide click features to jump from the design model to the referenced relevant information for a specific part; - provide BIM import facilities for thermal/imaging, acoustic/vibration, 3D laser scanning measurement data enhanced with 3D geometry (overlays for the BIM model) and energy performance data. The measurement instruments have their own native software which is able to provide export files containing the measurement data in a native format. - provide software for model checking and clash detection for models stored on the BIM server.	Regarding technical and functional requirements, the toolset should support converting and importing files from the applied measurement tools to a database to enhance the elements of the BIM. The toolset should also be able to connect several databases and to provide functionality to set up a dashboard with indicators.
3. **General requirements related to the practical use of the tool**	Workers use the toolset to be informed about daily activities and relevant related information. By doing this, workers can efficiently start and finish their work.	The toolset should be able to visualize planning information. Information may be related to activities to do, their duration, involved workers/managers, work conditions and required instruments. The toolset should also be able to apply authorization (user and user groups).	The toolset should be able to import planning data from an ERP software (enterprise resource planning).

4. General requirements regarding evaluation and decision-making processes	In the evaluation and decision making process, workers get back results from instruments or simulation and calculation software. Results, deviation thresholds and deviation's impact on KPIs will be shown in the interface. Based on the results, workers make decisions how to deal with deviations. Some cases require consulting managers in the decision making process. Inspection results, findings and decisions get updated in certain databases. All created evaluations results and all accessory decisions and reports will be stored in a certain database or updated to BIM.	To support functional requirements, the following technical requirements should be provided by the INSITER toolset: – information related to building performance during the inspection/realization process; – project managers provided with a KPI dashboard for the analysis of measurements data enabling him/her to take the right decisions, and – the BIM import facility should be able to import the native files and convert and store it into a database so that it can be used as source data for the KPI dashboard and visualization within the 3D model.	Regarding technical and functional requirements, the toolset should provide visualization of readings, calculation and simulation results against inspection related thresholds and KPIs. In some cases, depending on the possibility for data exchange with software packages, the toolset may provide automatically import/export data with those packages. This issues will extensively be studied in next steps of T3.2. The toolset should support data update to BIM model including readings, calculation, simulations performance results. By finishing inspection processes, the as-designed BIM model will be converted to the as-built BIM model.

List of acronyms and abbreviations

AR	: Augmented Reality
BCF	: BIM Collaboration Format
BIM	: Building Information Modelling
BMS	: Building Management System
CMIS	: Content Management Interoperability services
DoA	: Description Of Action
EeB	: Energy efficient Buildings
GUID	: Globally Unique IDentifier
HTML	: HyperText Markup Language
HVAC	: Heating, Ventilation, Air Conditioning
ICT	: Information and Communications Technology
IFC	: Industry Foundation Classes
KPI	: Key Performance Indicator
MEP	: Mechanical , Electrical, Plumbing technologies
XML	: Extensible Markup Language

Definitions

INSITER toolset

Refers to hardware, software and related servers and database to be developed for the purpose of this project.

Augmented reality

A live view of the real world where elements are augmented with graphics, video, sound or location data

Building Management System

A computer-based system for controlling and monitoring mechanical and electrical equipment of a building

Content Management Interoperability services

An open ICT standard for management of unstructured data within an organization

Hypertext Markup Language

A standard mark-up language for creating web pages

Key Performance Indicator

Key Performance Indicator is a quantifiable output measure, agreed upon beforehand, that can be used to estimate or compare performance of buildings or (sub)-systems

Extensible Markup Language

A text-based format used to share data on the World Wide Web, Intranets and elsewhere

Table of Contents

1. Introduction and scope

1.1 Task description and objectives

The main task for D3.3 is to determine the functional requirements and integration architecture for building quality and energy assessments. This will both support the assembly/construction workers carrying out the assessments on site and support the project manager with the analysis of measurements data hence providing the ability to take the right decisions. It will be applicable for different types of projects (new building, renovation/refurbishments and planned maintenance).

At a later stage, as part of task T3.2 and deliverable D3.4, a software prototype will be developed and implemented.

The objectives of this deliverable are:

- providing a good description of the functionalities the INSITER toolset to support self-inspection and self-instruction processes;
- providing a clear overview of already available software on the market. This will be done by reviewing the state-of-the-art software tools for the assessment of building conditions, energy and comfort assessment;
- providing initial concepts/ideas on how those assessment tools could be integrated for building quality and energy/comfort performance. The integration of those tools will be reported in D3.4;
- providing initial concepts/ideas on how interfaces between the assessment tools and the BIM server and the Building Maintenance Systems (BMS) could look like. Further development of those interfaces will be reported in D3.4, and;
- providing initial concepts/ideas on the embedding of key performance indicators, calculation, and measurement and verification protocols in the software tools. Further development of those concepts and relevant algorithms will be reported in D3.4.

1.2 Contribution of partners

This deliverable is created by DMO (WP and task-leader), DWA (lead author), FHGIPA, HVC and RDF.

Bi-weekly telco's and several physical meetings took place in the Netherlands, Germany and Bulgaria in order to collect information, have technical discussions and elaborate the software architecture concept.

Intensive collaboration with WP1 and WP4 partners took place in order to inventory the toolset functionalities based on the process requirements.

Most of the inputs for the requirements as stated in this document were collected from:

- WP1 (IAA, 3L, DWA) who described the process in D1.1, D1.4 and D1.6;
- WP2 (FHGIPA) for augmented reality;
- WP3 (RDF) for planning and cost software;
- WP4 (HVC, RDF) for BIM structure and architecture.

1.3 Relation to other activities in the project

The aim of INSITER is to develop new methodology for self-instruction and self-inspection activities during the realisation process. Both processes will support construction workers executing their work as efficient and effective as possible. Both processes will be supported by the INSITER toolset. The INSITER toolset is the whole of hardware, software, servers and databases supporting instructing, inspecting, measuring, evaluation and communication activities supporting the INSITER methodology. Figure 1.1 illustrates the INSITER toolset and its connections for both processes:

- Self-instruction process, supporting:
  - communication and visualisation of instructions & guidelines, to be developed in T1.1 & T4.2; and
  - communication and visualisation of planning and budget information, to be developed in T3.1;
- Self-inspection process, supporting:
  - communication and visualisation of inspection activities and related guidelines, to be developed in T1.1, T1.2 and T1.3;
  - data exchange between instruments (readings), BIM (building specifications), simulation packages (simulated building performance) and monitoring tools (simulated building performance), to be developed in WP4, WP2 and WP3;
  - decision making process regarding building errors and deviations, to be developed in T1.2 and T1.3.

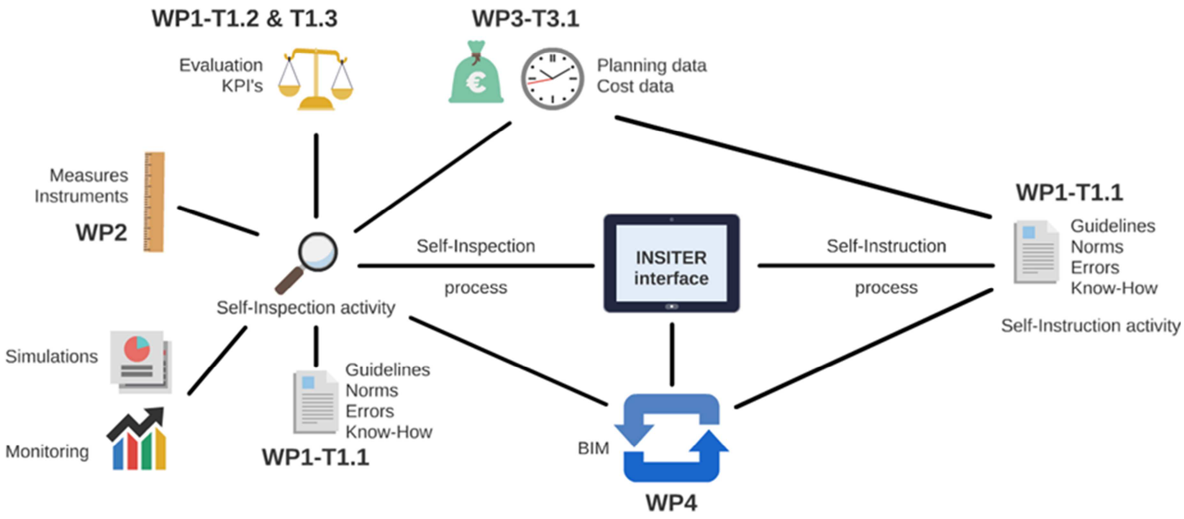
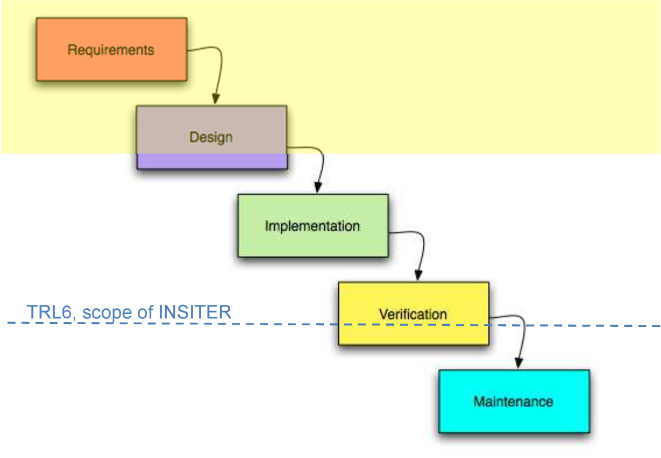


Figure 1.1: connections related to the INSITER toolset

The software development will be executed according to the widely acknowledged Waterfall model. This model exists of a set of process steps that are executed in a sequential order as is shown in the scheme below.

Waterfall model	Coverage of current and next deliverables
 Requirements Design Implementation TRL6, scope of INSITER Verification Maintenance	Main coverage of this deliverable D3.3
	Fundamental design covered in this deliverable
	Elaborated design and software prototyping will be the focus of next deliverable D3.4
	Software prototype TRL 6+ will be demonstrated in INSITER; the commercial development and implementation incl. maintenance for the client are part of future exploitation, yet out of scope of R&D in INSITER project

Note that the requirements collection and analysis is part of D3.3 and also a global idea of the functional design is part of D3.3. The actual functional design will be worked out in D3.4.

2. Requirements

This chapter inventories and describes functional, technical and operability requirements for the INSITER toolset as applicable for deliverable D3.3. In this chapter some essential issues have been considered: 1) analysis of already existing commercial tools, 2) challenges and needs of construction companies to proactively prevent construction errors, and 3) needs to assess construction errors in relation to key performance indicators.
The inventory is mainly based on both self-instruction and self-inspection processes as (to be) developed in WP1.
Other quality issues related to project management, including scheduling and costs, are a subject of Task 3.1 and will be reported in both D3.1 and D3.2.

2.1 Definition of requirements

In literature, a lot of definitions have been introduced regarding software requirements. Definitions depend on the study field and their type. To create a common understanding of what a requirement does mean in INSITER, it is very essential to have a clear definition and classification of requirements.

In the field of system engineering, a requirement is a necessary task, action or activity that must be accomplished by a software. In most cases, requirements are related to a condition or capability needed by a user or a system to solve a problem or achieve an objective. In the context of INSITER, the following ***definition*** will be used: '*requirements are conditions or capabilities that must be met or possessed by the INSITER toolset or its component to satisfy the self-instruction and self-inspection processes and all related tasks to be executed by the user during the stages of a building project"*. For practical consideration the wording "*tool"* and "*worker"* will be used further in this document.

Definition and ***description*** of requirements should consider some essential aspects:
1) requirements address *what* a system should do, but do not deal with how the system will address the function,
2) requirements address activities that constitute task objectives; but do not provide bells and whistles.

Quality of requirements[1] could be decided based on several characteristics:

- *necessary* (whether or not the requirement is required for proper working of the toolset without missing any of its capabilities),
- *concise* (whether or not the requirement is properly described to easily give its meaning and clarifying its intent),
- *feasible* (whether or not the requirement is achievable within the available resources of knowledge, time and money),
- *standalone* (whether or not the requirement can stand on its own and doesn't need any further amplification),
- *consistent* (whether or not a requirement contradicts with other requirements),
- *unambiguous* (whether or not a requirement can be interpret in only one way) and
- *verifiable* (whether or not can be determined if a requirement has been met).

[1] Gonzalez & Fowler, Developing Functional Requirements for ITS Projects, 2002.

In the literature, authors recommend avoiding certain words which are considered as either too general or too vague to suite quality requirements. Some of those words are: “flexible,” “fault tolerant,” “high fidelity,” “adaptable,” “rapid or fast,” “adequate,” “user friendly,” “support,” “maximize,” “minimize,” “and/or,” “etc.”, and “may.”

2.1.1 Levels of requirements

The INSITER toolset and processes will be used by workers from different types of companies and serving self-instruction and self-inspection processes during several stages of the realization process. Therefore, requirements for the toolset may be influenced by special needs of those companies. In the current version of this deliverable, an inventory of all types and levels of requirements is shown, considering pre-defined levels. Levels of requirements determine the essence of having such a requirement. These levels have been defined as follows:

- ***Must have*** requirements: toolset objectives cannot be met without these requirements.
- ***Should have*** requirements: toolset objectives can be met without these requirements but not as well as with it.
- ***Could have*** requirements: toolset requirements only map to one or more principles.
- ***Wish list*** requirements: these requirement do not map to any project objectives or principles.

2.1.2 Types of requirements

This document focuses on functional requirements. However, functional requirements may be categorized by several levels according to their function type. In this context we list three types of functional requirements:

- **Functional requirements**
  Defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs.
- **Technical requirements**
  A technical requirement refers to the technical aspects that a system must fulfil
- **Interoperability requirements**
  Defines how technical systems or components are shared or connected to each other

2.2 Analysis of existing software tools

Several inspection management tools already exist in the market. These tools aim at supporting inspectors and project managers in building quality issues. The tools focus on efficiently and effectively executing construction activities and ensuring project management criteria (quality, cost and schedule). In this section two tools are discussed: **BIM 360 Field** (as most powerful commercial tool) and **ED-Controls** (as most used tool in the Netherlands and due to the experience of DWA).

2.2.1 BIM 360 Field

BIM 360™ Field[2] is field management software for 2D and 3D environments that combines mobile technologies at the construction site with cloud-based collaboration and reporting. BIM 360 Field provides field workers with critical information to improve quality, safety, and commissioning for construction projects.

The tool provides several functionalities including:

- managing quality by using standardized checklist templates that are easy to use and customize;
- using pushpins to quickly communicate project's issues and status;
- providing overview of safety performance metrics to help proactively avoid incidents;
- capturing system and equipment information electronically to help save time and reduce errors;
- accessing and viewing photos, operation and maintenance manuals, manufacturer cut sheets;
- providing the team with up-to-date project status information;
- using standard reports out of the box and schedule reports to automatically send to all project team members and subcontractors;
- locating equipment in a model automatically using your iPad, and use pushpins in 3D to mark issues;
- mapping equipment data updates to the model (BIM 360 Glue and Navisworks or Revit BIM software); and
- providing cross-project reporting.

[2] http://www.autodesk.com/products/bim-360-field/overview

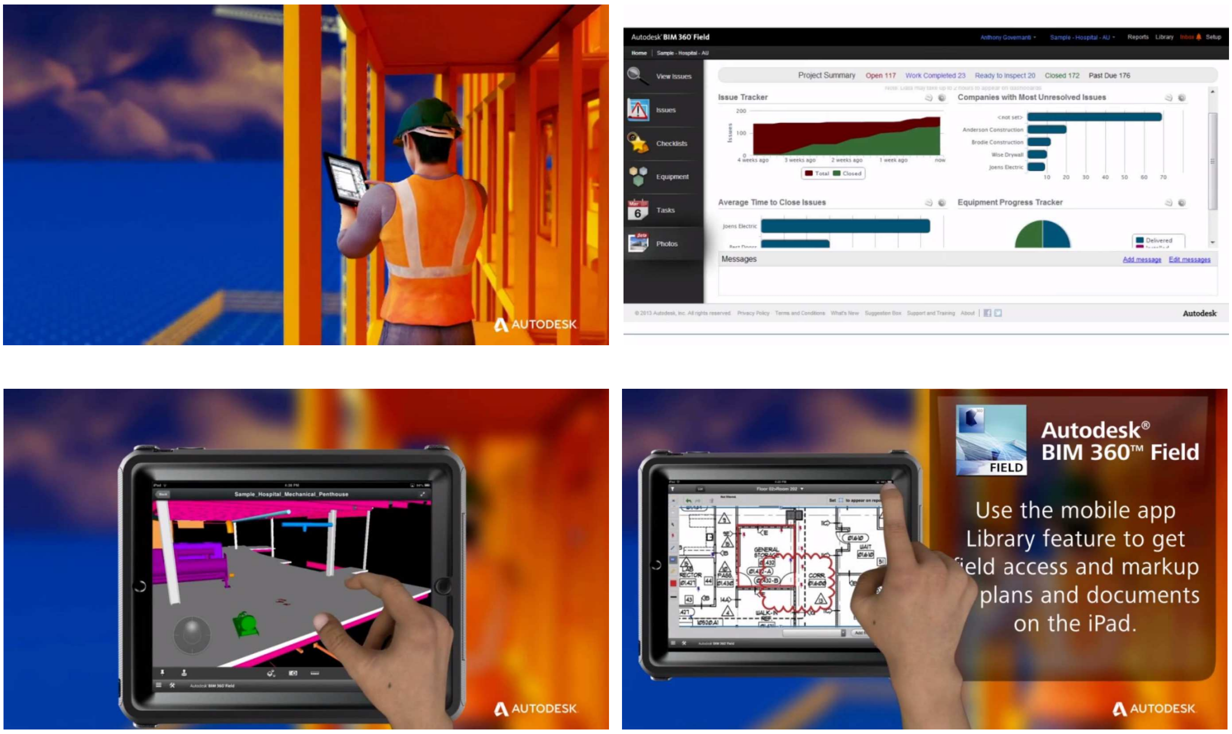

AUTODESK
Autodesk BIM 360 Field
View Issues
Issues
Checklists
Equipment
Tasks
Photos
Project Summary Open 117 Work Completed 23 Ready to inspect 20 Closed 172 Past Due 176
Issue Tracker
Companies with Most Unresolved Issues
Average Time to Close Issues
Equipment Progress Tracker
Messages
Autodesk
AUTODESK
Autodesk® BIM 360™ Field
FIELD
Use the mobile app Library feature to get field access and markup plans and documents on the iPad.
AUTODESK

Figure 2.1: BIM 360 Field

BIM 360 Field is a powerful inspection management tool. However, it does not support self-instruction and self-inspection processes. Additionally, it does not support decision support processes for building quality and energy assessment. The communication and connectivity to BIM requires BIM 360 Glue; an additional tool from AutoDesk (additional license).

2.2.2 ED-Controls

ED controls[3] is a field management tool during the construction process. The tool aims at providing an overview of executed inspections and construction errors still need to be solved while supporting communication activities between all project partners.

The tool provides several functionalities including:

- login both on a portable device as web based;
- defining roles and responsibilities including administrators, project managers and inspectors;
- creating a project and adjusting project settings,
- uploading drawings;
- editing pictures;
- creating and importing audits forms for several types of inspection activities;
- supporting ticketing and mailing possibilities;

[3] www.edcontrols.com/en

- web-based helpdesk.

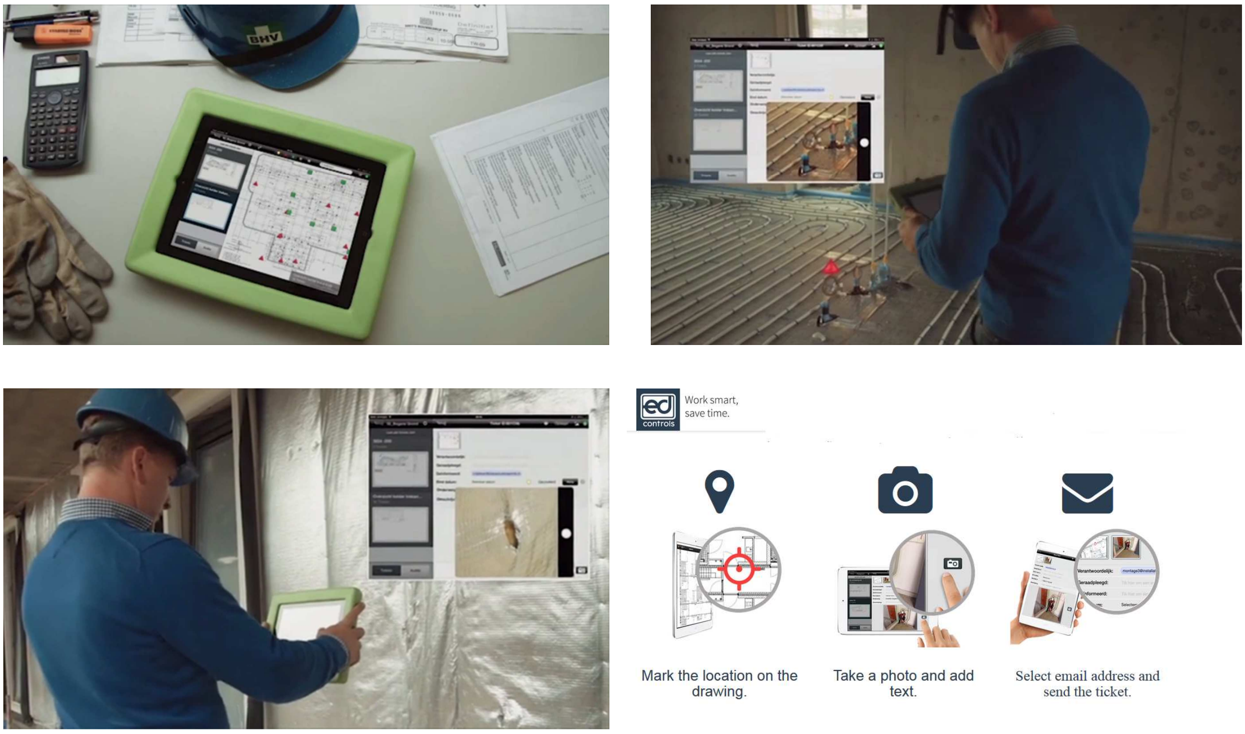

Work smart, save time.
Mark the location on the drawing.
Take a photo and add text.
Select email address and send the ticket.

Figure 2.2: ED-Controls

The current commercial version of ED-controls doesn't support BIM connectivity. However, it uses an IFC file for creating 2D drawings and for GPS coordination. ED-controls is a management tool for inspection activities but it doesn't support any kind of building quality and energy assessment functionalities. The tool doesn't support self-instruction and self-inspection processes. The tool doesn't provide a GPS coordination functionality. However, location can be traced by scanning (Bar Code) of already installed components and systems (already linked to BIM).

2.2.3 Shortcomings of existing tools

As discussed above, existing inspection management tools aim at supporting inspection activities by the project managers and inspectors. Tools provide a powerful tool to ensure construction quality issues (following project schedule and related norms) and communication regarding (what work-to-complete and by whom).

The most remarkable shortcomings of existing tools are:

- Tools register and manage inspection processes instead of preventing them.
- Tools do not support the assessment of construction errors in the realization phase.

2.3 INSITER R&D challenges

2.3.1 Building quality and energy assessment

Construction projects become more complicated. Building owners have higher expectations and more needs related to the use of buildings, sustainability, energy consumption, thermal comfort, image and satisfaction of building users. These issues requests that building should meet high level of quality issues in the design/realization phase as well as in the use phase.

Optimizing building design in the very first phases of the project could be ensured by understanding owner's needs/expectations and translate them into right building design, measurements and suitable HVAC/MEP systems. However, the realization phase becomes a very critical step in the ensuring building quality as frequently errors happen due to lack of powerful tools that can support the execution/inspection processes as well as the decision making process during the realization phase. Construction managers, and also workers, need to be supported in the evaluation of building errors and the assessment of how to deal with.

Existing commercial tools try to fulfil these needs. As discussed in in previous sections, existing commercial tools do provide powerful tools for inspection activities in the realization phase. However, shortcomings regarding assessment of building quality and energy performance still exist. Also proactively preventing construction errors, means self-instruction and self-inspection activities, is still not covered by the current tools.

The INSITER project improves current inspection tools by providing two essential functionalities:

- Pro-actively preventing errors by means of self-instruction and self-inspection methodologies; and
- Supporting the decision making process in assessing construction errors.

The main challenges for the INSITER project are:

- Developing a new self-instruction methodology and related ICT-supported tools to prevent construction errors;
- Developing a new self-inspection methodology and related ICT-supported tools to inspect quality by workers themselves;
- Improving existing measuring instruments;
- Improving existing software; and
- Developing the ICT architecture that facilitates the above mentioned processes.

2.4 Targeted final achievement

INSITER aims at developing a user friendly software toolset that supports, on the one hand, efficiently and effectively execution of assembly/construction activities by workers and carrying out the assessment of building quality, on the other hand. The toolset will also support project managers with the analysis and evaluation of measurements data via a KPI dashboard (described in the next chapters). The requirements for the toolset are 1) based on the analysis in Section 2.2, 2) derived from self-instruction and self-inspection processes, 3) other work package deliverables and use cases from work package 5. The software itself is deliverable D3.4.

2.5 Self-instruction process

The self-instruction process aims at providing assembly/construction workers with information to prevent construction errors (product specifications, process description, planning data and relevant know-how information). It also aims at preventing errors during the realization phase by carrying out work 'first time right'. Figure 2.3 illustrates which information could be provided by the INSITER toolset to the construction/assembly worker.

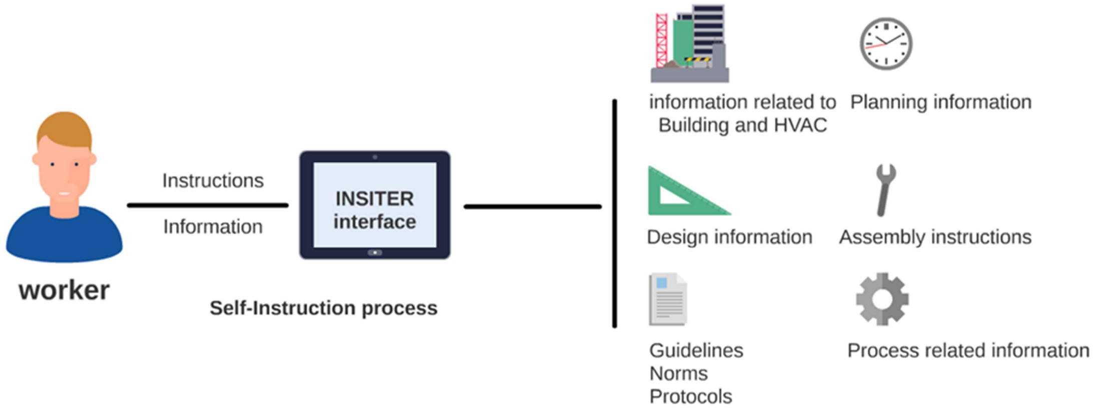


Figure 2.3: information sources to be supported by the INSITER toolset

2.5.1 Concept of self-instruction process

In the self-instruction process, assembly workers get the possibility to access data related to the work to be done. These data could be related to building design information, specification of building elements and technical equipment, related norms and lists of most common errors. Assembly workers may also benefit from viewing drawings and work instruction related to the work to be done. Figure 2.4 illustrate the self-instruction process.

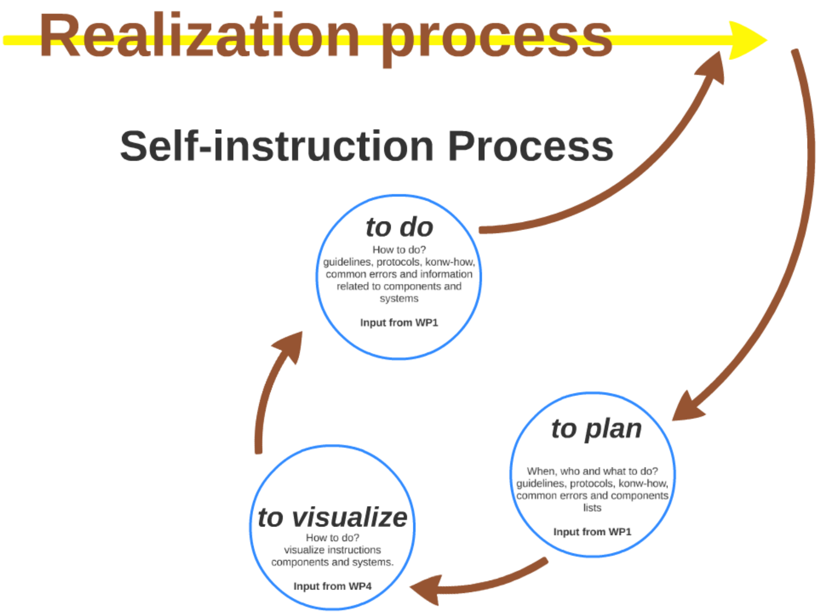

Realization process
Self-instruction Process
to do
How to do?
guidelines, protocols, konw-how, common errors and information related to components and systems
Input from WP1
to plan
When, who and what to do?
guidelines, protocols, konw-how, common errors and components lists
Input from WP1
to visualize
How to do?
visualize instructions components and systems.
Input from WP4

Figure 2.4: information sources to be supported by the INSITER toolset

The INSITER toolset will thus support the following functionalities:

- providing access to process related information (daily activities, planning);
- providing access to project and process related information (building design, HVAC design);
- providing access to product related information (product specification);
- providing access to assembly & construction related information (how to assembly or to install);
- visualizing above mentioned information and documents.

2.5.2 Requirements related to self-instruction processes

As described in Section 2.2, the INSITER toolset will support construction/assembly workers in their activities through a self-instruction process. This process will be extensively reported in WP1 (D1.2 and D1.3) and WP4 (D4.4). In the following sections self-inspection requirements will be discussed.

Functional requirements

Workers start their daily work by following project planning and executing assigned activities (as discussed in previous section). Workers benefit from information about construction/assembly activities when to perform (process), how to perform (guidelines and know-how) but also what to perform (product and building specifications).

Technical requirements

The toolset should be able:

- to show relevant instructions and information related to the assigned activities. This can be done by web-based application which can be used device independent;
- to show available documents on a web page and to offer the possibility to download documents, images and movies that are related to a certain part of the BIM model via the Internet.

Interoperability requirements

The toolset should be able to import product specification (database or BIM), building & design specification (BIM) and related guidelines (database). The tools should also be able to import documents and checklists (mostly related to standard processes).

2.6 Self-inspection process

Building quality and energy performance depend on the provided quality in the different stages of the construction process; starting from an optimized design to perfectly executed realization activities. In each stage of this process, inspection activities could be carried out. The toolset will help the end-users to deal with questions like: Does the design ensure excepted quality levels? Do assembly activities ensure right construction of the building and its systems?
In this section, the self-inspection process will be described based on the proposal in the DoA.

2.6.1 Concept of self-inspection process

To ensure good performing buildings, the following issues should be considered:

1. optimized building design that:
   - fits owner's needs & expectations,
   - fits legal requirements,
   - meets sustainability & energy consumption goals, and
   - provides healthy, safe and productive indoor environment.
2. right building components and elements have been purchased,
3. components and elements have been transported and delivered to the site (undamaged);
4. building elements have been correctly assembled;
5. HVAC/MEP components/systems have been correctly installed; and
6. building and HVAC/MEP settings have been correctly installed.

The INSITER project does consider an optimized design. For this reason, the first step will not be discussed in the following sections. The inspection process will consider only steps from 2 to 6.

2.6.2 Purchased, transported and delivered goods

One of the common errors in this phase of the project is that purchased goods deviate from the design specifications (different of comparable goods purchased). These deviations may influence building performance. Using model checking software may avoid this kind of errors. This first inspection step should, therefore, ensure that the right components and elements have been purchased (Figure 2.5).
Another common error is that goods could be damaged during transport and delivery. Damaged elements/components may perform poorly. The aim of this related inspection process is to ensure that goods are not damaged.
To support this stage, the INSITER toolset will support the following functionalities:

- model checking and clash detection (Model Checking is to check BIM models for rule-based conflicts or specific values; Clash Detection is to check the design (or re-design) for geometrical conflicts);
- product detection possibility (e.g. QR scanner); and
- possibility for inspecting damaged goods through a visual inspection process.

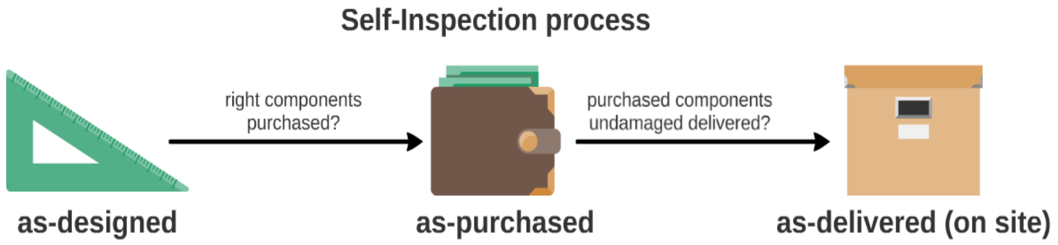

Self-Inspection process
right components purchased?
purchased components undamaged delivered?
as-designed
as-purchased
as-delivered (on site)

Figure 2.5: self-inspection process of purchased goods

2.6.3 On-site assembly

The main self-inspection process in the INSITER project is based on the plan-do-check-act process. Figure 2.3 illustrates this process.

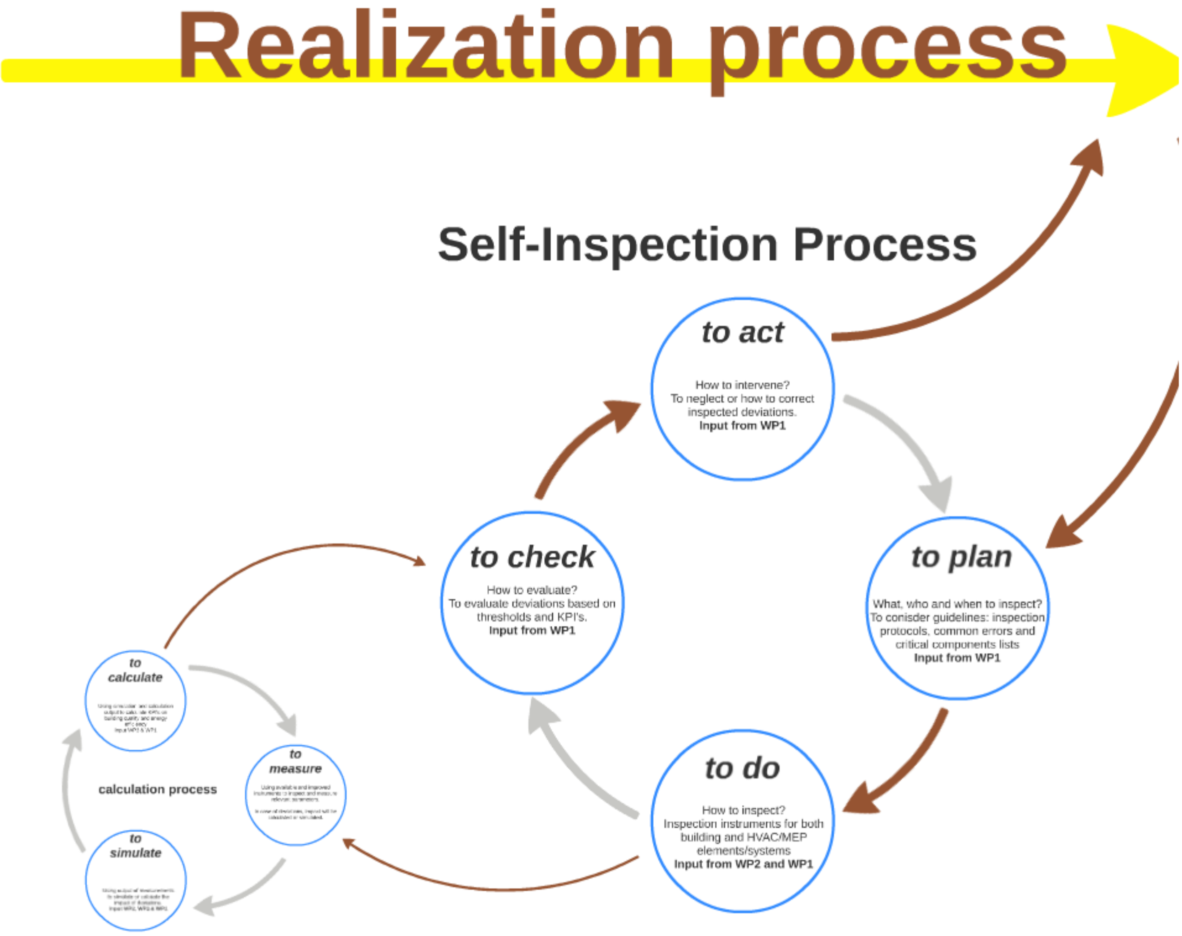

Realization process
Self-Inspection Process
to act
How to intervene?
To neglect or how to correct inspected deviations.
Input from WP1
to plan
What, who and when to inspect?
To conisder guidelines: inspection protocols, common errors and critical components lists
Input from WP1
to do
How to inspect?
Inspection instruments for both building and HVAC/MEP elements/systems
Input from WP2 and WP1
to check
How to evaluate?
To evaluate deviations based on thresholds and KPI's.
Input from WP1
to calculate
Using simulation and calculation output to calculate KPI's on building quality and energy efficiency
Input WP3 & WP1
to measure
Using available and improved instruments to inspect and measure relevant parameters.
In case of deviations, impact will be calculated or simulated.
to simulate
Using output of measurements to simulate or calculate the impact of deviations.
Input WP2, WP3 & WP1
calculation process

Figure 2.6: conceptual illustration of the self-inspection process

In the ***plan stage*** of the self-inspection process, construction/assembly workers may benefit from access to relevant schedule/activities information including what and when activities/inspections should be carried out and by who (access to planning information). Information may also contain guidelines regarding those activities, most common errors related to the work to be carried, relevant norms and instruments to be used. These data will be developed in WP1 (T1.1, T1.2 and T1.3).

To support this stage, the INSITER toolset will support the following functionalities:

- access to project planning and activity database;
- access to BIM;
- access to several databases e.g. guidelines, errors and norms;
- visualization of above data/information.

In the ***do stage*** of the self-inspection process, construction/assembly workers may benefit from access to information related to guidelines of instruments, relevant know-how information, common errors and applicable protocols. These data/information support workers to carry out inspection as efficient and effective as possible.

To support this stage, the following functionalities should be supported by the INSITER toolset:

- access to several databases e.g. instruments, their relevant protocols and readings;
- import/export instrument readings; and
- repetitively comparing the reality on site with the design model by means of Augmented Reality (AR) / BIM,

In the ***check stage*** of self-inspection process, construction/assembly workers should be supported in evaluating the impact of deviations on building performance and energy efficiency (or KPIs). Based on the outcome of this stage, the worker or his/her manager will be able to assess how to deal with that deviation. This part of the process may be carried out in different ways depending on the inspection in question. In some cases, the evaluation process could be performed by means of simple calculations. In other cases, the evaluation may go through a complicated simulation process. Figure 2.4 illustrates how this evaluation process may look like. Generally speaking, the evaluation process goes, as part of do-check process, through the following sub-stages:

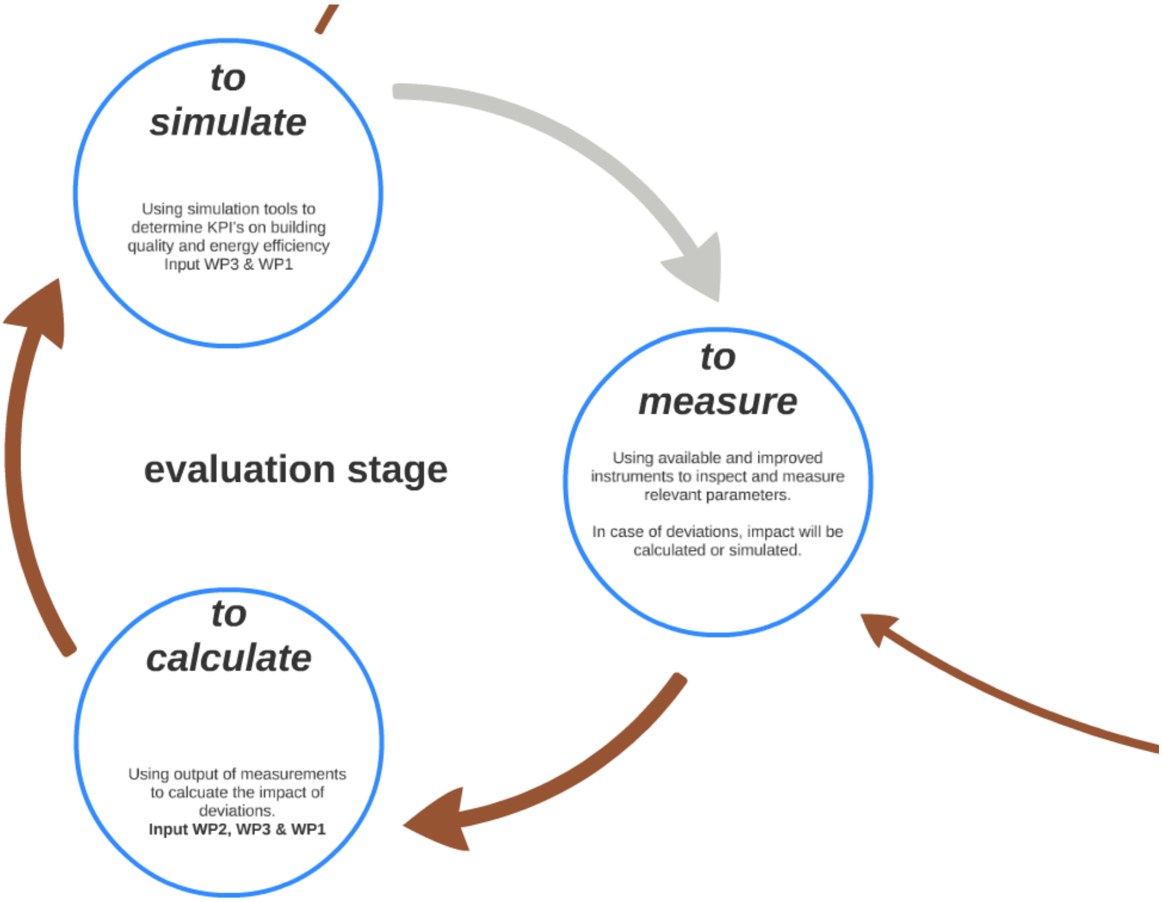

to simulate
Using simulation tools to determine KPI's on building quality and energy efficiency Input WP3 & WP1
to measure
Using available and improved instruments to inspect and measure relevant parameters.
In case of deviations, impact will be calculated or simulated.
evaluation stage
to calculate
Using output of measurements to calcuate the impact of deviations.
Input WP2, WP3 & WP1

Figure 2.7: conceptual illustration of evaluation stage in the self-inspection process

In the ***measure and simulate stages***, as part of the ***do stage***, the worker may find some deviation in the readings. The impact of this deviation on KPIs should be decided. In INSITER we consider three levels of evaluation:

- by means of thresholds: INSITER will also provide, for some specific cases, evaluation thresholds. These thresholds provide an easy way to make decision on deviations in the realization process. Thresholds are max/min values whereby deviations could be accepted or rejected. These values are based on international protocols and field experience.
- by means of calculation: in some cases only simple calculations are necessary to evaluate deviations in the realization process.
- in the most advance cases, INSITER will consult simulation or monitoring software to evaluate the impact of deviations in the realization process. However, this may be a complex and time consuming process.

At the end of this stage, KPIs could be determined and be shown to the worker/manager in order to decide how to deal with that deviation.

To support the above mentioned stages, the INSITER toolset should support the following functionalities:

- provide access to threshold database;
- export readings to calculation, simulation and monitoring software packages;

- import and visualize output from calculation and simulation software packages;
- possibility to overlay BIM with instrument readings and simulation outcomes by means of AR;
- visualization of KPIs for both the worker as well as the manager.

In the ***act stage***, findings of the check stage will be presented in relation to KPIs and thresholds. The impact on the agreed KPIs will decide how to deal with the deviation in question. In the most cases, workers should be able to assess the deviation by themselves. However, in some cases workers may discuss deviations with their managers. The INSITER self-inspection process assures the following possibilities:

- in case of negligible impact, deviations could be accepted and reported;
- in case of remarkable impact, deviations may be, in accordance with managers, accepted due to huge complications to delay or budget overrun. In this case, new performance results should be reported;
- in case of inadmissible impact, deviations/errors should be corrected and improved. In this case, workers may benefit from instruction how to improve.

To support these functionalities, the INSITER toolset should support:

- presenting and visualizing the impact of deviation in question in relation to KPIs and threshold,
- communication between workers and their managers in order to assess deviations;
- reporting the whole inspection process and saving relevant data in related databases; and
- updating the BIM model with new values related to inspections, deviations and impact on KPIs.

2.6.4 Requirements related to self-inspection processes

Functional requirements

As described in the Section 2.3 workers use the INSITER toolset to perform inspections by themselves. The toolset will act as a connecting platform for the several elements (instruments, sensors) involved in the several stages (do, check and act). In inspection processes, workers use the toolset to register defects and insert/import readings or measurements. They use the toolset to perform model checks and clash detections. In some cases, workers use the toolset to export readings to calculation simulation software to decide on the impact of found deviations on KPIs. The export process may vary from automatically (connection though IT solutions) to manually (self-entry) data export. For this reason, the evaluation and decision-making process are discussed separately in Section 2.7.

Technical requirements

The toolset should be able to:

- provide workers with the ability to register defects;
- provide the assembly/construction with the ability to repetitively verify the construction during realization against the design model;
- visualize the design model on-site including HAVC/MEP systems;
- provide augmented reality (web-based application) overlaying the reality on site with (parts of) the design model;
- provide workers with drawings and other documents;
- provide click features to jump from the design model to the referenced relevant information for a specific part;
- register defects and quality issues;
- provide BIM import facilities for thermal/imaging, acoustic/vibration, 3D laser scanning measurement data enhanced with 3D geometry (overlays for the BIM model) and energy performance data. The measurement instruments have their own native software which is able to provide export files containing the measurement data in a native format.
- provide software for model checking and clash detection for models stored on the BIM server.

Interoperability requirements

Regarding technical and functional requirements, the toolset should support converting and importing files from the applied measurement tools to a database to enhance the elements of the BIM. The toolset should also be able to connect several databases and to provide functionality to set up a dashboard with indicators.

The choice of database is under investigation. The decision/clarification so far comprises: 3 different databases are needed; Open Source BIM server (for BIM/IFC models) and Microsoft SharePoint (for all other files) will be used; and the third database will be used for connecting instances in the BIM model and the different files + other semantic data.

Further investigation within the framework of software development will address the following research questions: Do we use SQL or non-SQL databases? In case of SQL, it will be a PostgreSQL database. In case of non-SQL it will be Mongo-DB or Couch DB. From INSITER consortium’s point of view: PostgreSQL => CARTIF has deep knowledge about this (a test server is running); Mongo-DB => DWA/Simaxx has deep knowledge about this; Couch DB => Fraunhover has deep knowledge about this.

2.7 General requirements related to the practical use of the tool

This paragraph describes the General software requirements regarding practical use of the tool. A requirement is something that is essential to facilitate a certain process or to achieve a specific goal. When developing software these requirements are subdivided in three categories. These categories are explained below

In order to properly execute both self-instruction and self-inspection process, the toolset should create some basic functionalities to be used in the most available commercial tablets. These functionalities may be: log in with own password, text editing, mail editing, taking/editing photo's, GPS function, viewing serval types of documents (PDF, DOC, XLS, JPG…), print screen, Wi-Fi, 3G. We only focus on functionalities related to building quality and energy efficiency. However, the following requirement should also be mentioned.

Functional requirements

Workers use the toolset to be informed about daily activities and relevant related information. By doing this, workers can efficiently start and finish their work.

Technical requirements

The toolset should be able to visualize planning information. Information may be related to activities to do, their duration, involved workers/managers, work conditions and required instruments. The toolset should also be able to apply authorization (user and user groups).

Interoperability requirements

The toolset should be able to import planning data from an ERP software (enterprise resource planning).

2.8 General requirements regarding evaluation and decision-making processes

Functional requirements

In the evaluation and decision making process, workers get back results from instruments or simulation and calculation software. Results, deviation thresholds and deviation's impact on KPIs will be shown in the interface. Based on the results, workers make decisions how to deal with deviations. Some cases require consulting managers in the decision making process. Inspection results, findings and decisions get updated in certain databases. All created evaluations results and all accessory decisions and reports will be stored in a certain database or updated to BIM.

Technical requirements

To support functional requirements, the following technical requirements should be provided by the INSITER toolset:

- information related to building performance during the inspection/realization process;
- project managers provided with a KPI dashboard for the analysis of measurements data enabling him/her to take the right decisions, and
- the BIM import facility should be able to import the native files and convert and store it into a database so that it can be used as source data for the KPI dashboard and visualization within the 3D model.

Interoperability requirements

Regarding technical and functional requirements, the toolset should provide visualization of readings, calculation and simulation results against inspection related thresholds and KPIs. In some cases, depending on the possibility for data exchange with software packages, the toolset may provide automatically import/export data with those packages. This issues will extensively be studied in next steps of T3.2. The toolset should support data update to BIM model including readings, calculation, simulations performance results. By finishing inspection processes, the as-designed BIM model will be converted to the as-built BIM model.

3. Clustering requirements

In the previous chapter the functional requirements, technical requirements and interface requirements are described based on the self-instruction and self-inspection processes. Each process generates its own technical and interface requirements, in this chapter we will cluster the identified requirements in groups. Only the most necessary requirements will be clustered and further discussed.

The decision is based on the relevance of the requirements and the scope of the project. In other words, some other requirements mentioned in Chapter 2 will not be further discussed. Based on the requirements we have defined the following clusters:

- BIM related requirements
- Self-instruction requirements
- Self-inspection requirements
- Visualization related requirements
- Conversion related requirements

This deliverable focusses mainly on building quality and energy efficiency. A lot of those found requirements already have been mentioned in D3.1. Therefore, those requirements will not be mentioned here.

3.1 Functional requirement clustering

BIM Requirement	Brief description	Further details in
IFC upload/download	IFC 2x3 files upload and download functionality for the BIM server is required to feed the BIM server with new or updated IFC models and for exporting these models to other applications (like CAD)	D4.3
IFC model check	Identify conflicts in attributes of the IFC models to guarantee smooth BIM workflows	D4.3
Support for combined BIM and process data	The BIM server combines product, process and project data.	D3.2
Clash detection and rule based model checking	Identify geometrical and design rule based model conflicts during the design phase. Identify conflicts in the IFC models before they are identified in the field when it is too late.	D3.2

Data upload to BIM including: readings, measurements, calculation & simulation results, inspection evaluation and decision reports.	The BIM server is able to store geometrical data as well as measurement data, relevant documents, pictures.	D3.2
Self-instruction Requirement	**Brief description**	**Further details in**
Being able to provide info related to activities to do by workers	Instruction documents, movies, pictures can be provided to the worker	D1.2, D1.3, D4.3, and D3.4
Self-inspection Requirement	**Brief description**	**Further details in**
Being able to provide info related to how to inspect already done activities	For the part to be inspected all product, process and project related information is retrievable on site	D1.4, D1.5, D1.6 and D1.7
Being able to register defects/deviations	All product/process related deviations can be reported to the BIM server	D3.4
Being able to insert/import readings and measurements	Readings and measurement data is collected, converted and uploaded to the BIM server	D3.4, D1.4, D1.5, D1.6 and D1.7
Support communication between workers and managers	Facilitating communication processes for checking and planning	D3.4
Being able to insert/import calculation and simulation results	Calculation and simulation results from energy performance systems are uploaded to the BIM server	D3.4
Visualization Requirement	**Brief description**	**Further details in**
Being able to visualize info related to self-instructions	The worker on site is able to view self-instruction documents, movies and pictures on his mobile device	D3.2
Being able to visualize info related to self-inspection	The worker on site is able to view all information on his mobile device that may help in carrying out the inspection task	D3.2
Being able to visualize measurements and readings	The worker on site is able to view the measurements and readings that are available on the BIM server	D3.2
Being able to visualize calculation and simulation results	The worker on site is able to view the energy performance calculation and simulation results that are available on the BIM server	D3.2
Being able to visualize KPIs	The worker on site is able to view the actual status of the KPIs	D3.2

3.2 Technical requirement clustering

In the table below the technical requirements and related software applications are listed in order to fulfil the functional requirements as stated in 2.2 Chapter 6 describes the architecture and the necessary applications. The technical requirements of the needed application are described in the table below.

Software application	Technical requirements	Remarks
1. File sharing system	Web-based application which can be used device independent.	CMIS on SharePoint could be applied for this purpose.
	The toolset must be able to show the available documents on a web page and offers the possibility to download documents, images and movies that are related to a certain part of the BIM model via the internet .	The BIM model (part) will be provided with references (URL with GUID) to the documents on the file share system and the links will be executed when the user clicks on that part of the model. A HTML form on the web site is shown with the list of available documents for the BIM part.
	The toolset has search and navigation functionality.	
	The toolset applies authorization (user and user groups).	
2. Defect and quality issue registration tool	Web-based application which can be used device independent.	
	The toolset presents forms for defects and quality issues as will be defined by WP1.	
	A reference (URL) to the form is created automatically on the relevant BIM server part.	The BIM server parts are shown via the AR toolset and the HTML page where the BIM server part refers to contain a link to the defect and quality issue registration toolset which will be loaded with the relevant defects and quality issues for this part when the user clicks the link.
3. AR tool	Web-based application.	To be developed, reported and delivered by WP2.
	The application will be able to retrieve the 3D BIM model geometries and to visualize it.	
	The application will be able to overlay the reality as seen by the camera on a mobile device.	The computer vision based tracking establishes a correlation of the BIM-model and the real environment captured with the camera image via model and marker based tracking solutions
	The user may click on a BIM model part and is forwarded to a HTML web page that contains information about that part (documents and a link to start up the defect and quality issue tool).	The AR software could also use a predefined link with the BIM model part GUID as parameter.

4. Dashboard for KPIs and decision support	The application should be able to connect to several databases and to provide functionality to set up a dashboard with indicators.	The RE Suite dashboard will be applied for this purpose.
5. Import facility BIM server	Files from the applied measurement tools should be converted and imported into the BIM server enhanced with 3D geometry.	This may require middleware software to convert files in native format from the measurement devices into a more standard format that can be processed (like XML). See also adaptation layer BIM architecture D3.1 chapter 6.
5. Condition assessment	Web-based application which can be used device independent.	

3.3 Interoperability requirement clustering

Device related interoperability requirement	Brief description	Further details in
Being able to convert data files from measurement instruments to BIM data	The measurement devices produce data that is registered in a native format. These data should be converted into a format that can be stored in the BIM server for accessibility by the worker on site via his mobile device	D3.2
Planning and Cost related interoperability Requirement	**Brief description**	**Further details in**
Being able to convert planning and cost data to BIM data	The data coming from planning and cost calculation tools are registered in a native format. This data should be converted into a format that can be stored in the BIM server for accessibility by the worker on site via his mobile device	D3.2
IFC related interoperability Requirement	**Brief description**	**Further details in**
Being able to import and export IFC files from and to other systems	IFC files can be imported and exported from/to other systems	D3.2

4. Software functionalities (features)

As described in the first chapter, before a software application is going to be coded (writing the source code), a functional design is needed. This detailed functional design (which is a part the next deliverable), will be generated based on a collection of the main required functionalities (these main functionalities are described in this chapter). Accordingly, Chapter 7 will give an impression how this will look like in the software itself by describing this so called proof-of-concept.

4.1 File Sharing System

4.1.1 Functionality

A file sharing system is required within the INSITER toolset in order to share documents between the worker on site and the back office. The file sharing system is provided with project, product and process related documents so that the worker on site is able to download the documents on his/her mobile device.

The system consists basically of two entities:

1. File Sharing System as backend software application that runs on a server
2. A mobile App that will be applied by the worker on site to fetch the required documents from the SharePoint server

The project manager takes care that the file sharing system is provided with all required documents. This requires a certain structure in which the documents are organized. The structure should be created applying document libraries in the following way.

Process library:
In this library the generic process related documents are stored like process flow, quality assurance documents

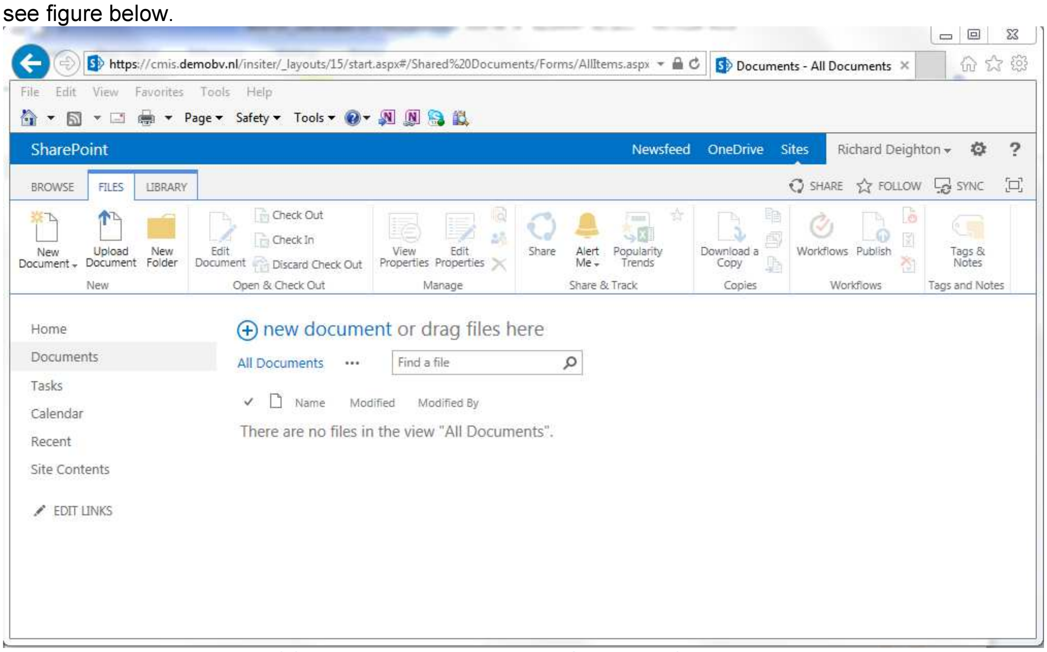
Product library:
In this library the product related libraries are stored like panels, glazing, building materials, equipment.

Project library:
In this library the project related documents are stored like contracts, project organization, goals, planning, progress, quality control documents.

All document storage locations are provided with unique identifiers so they can be uniquely addressed by means of a URL (link). This way documents that are related to an IFC element from the BIM model can easily be found applying the unique identifier of the IFC element itself (GUID)

4.1.2 Providing documents to the file sharing system

The back office workers apply their web browser in order to store documents, pictures, movies on the file sharing system see figure below.



There are no files in the view "All Documents".

Figure 4.1: illustration of file sharing system in INSITER software

A more detailed description of the workflow is provided by the UML sequence diagram below outlining the interaction e.g. between the project manager and the file sharing system.

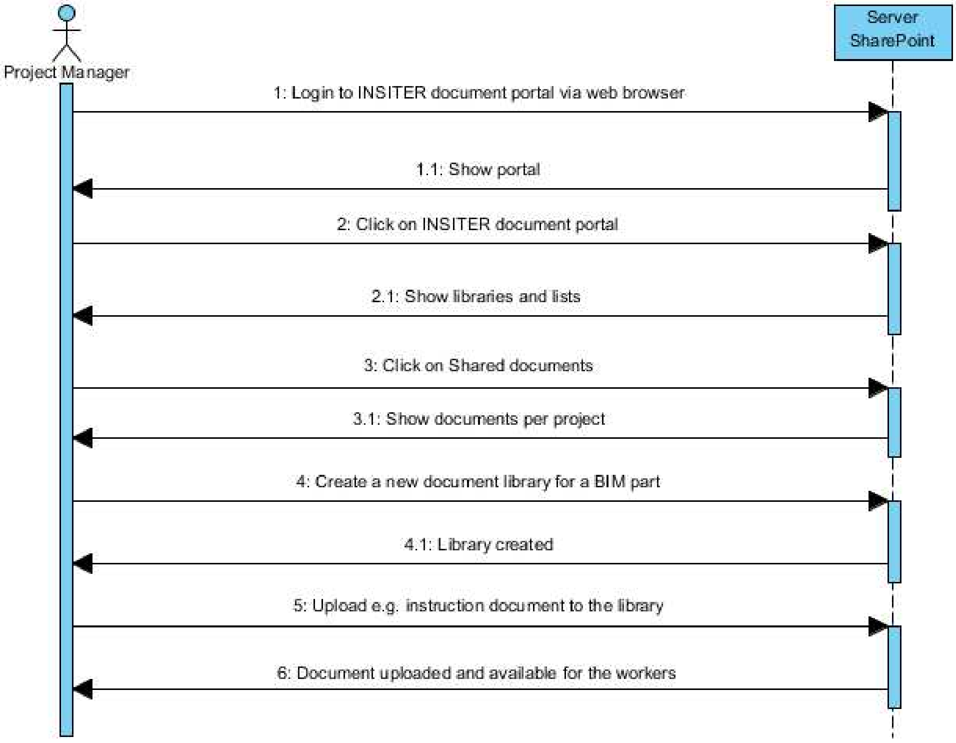

Project Manager
Server SharePoint
1: Login to INSITER document portal via web browser
1.1: Show portal
2: Click on INSITER document portal
2.1: Show libraries and lists
3: Click on Shared documents
3.1: Show documents per project
4: Create a new document library for a BIM part
4.1: Library created
5: Upload e.g. instruction document to the library
6: Document uploaded and available for the workers

Figure 4.2: UML sequence diagram between project manager and file sharing in INSITER software

4.1.3 Retrieving documents from the file sharing system

The worker on site is able to fetch the documents from the SharePoint server via an App on his/her mobile device. Documents are presented on the mobile device as shown in the picture below

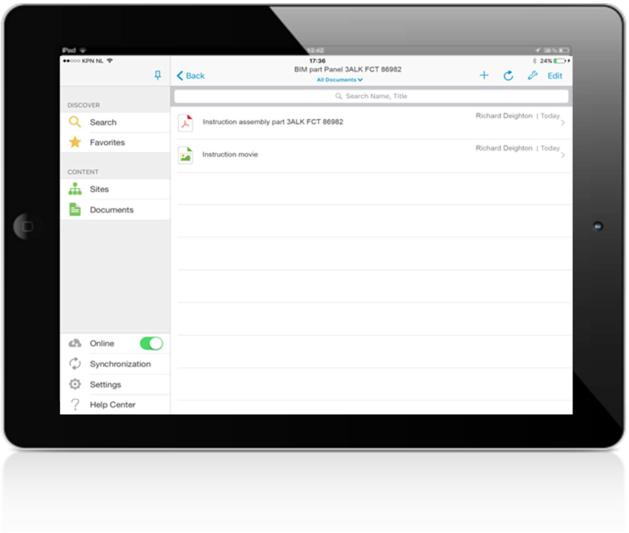


Figure 4.3: illustration of document retrieval on mobile device

A more detailed description of the workflow is provided by the UML sequence diagram below outlining the interactions between the worker on site, the mobile App and the SharePoint server.

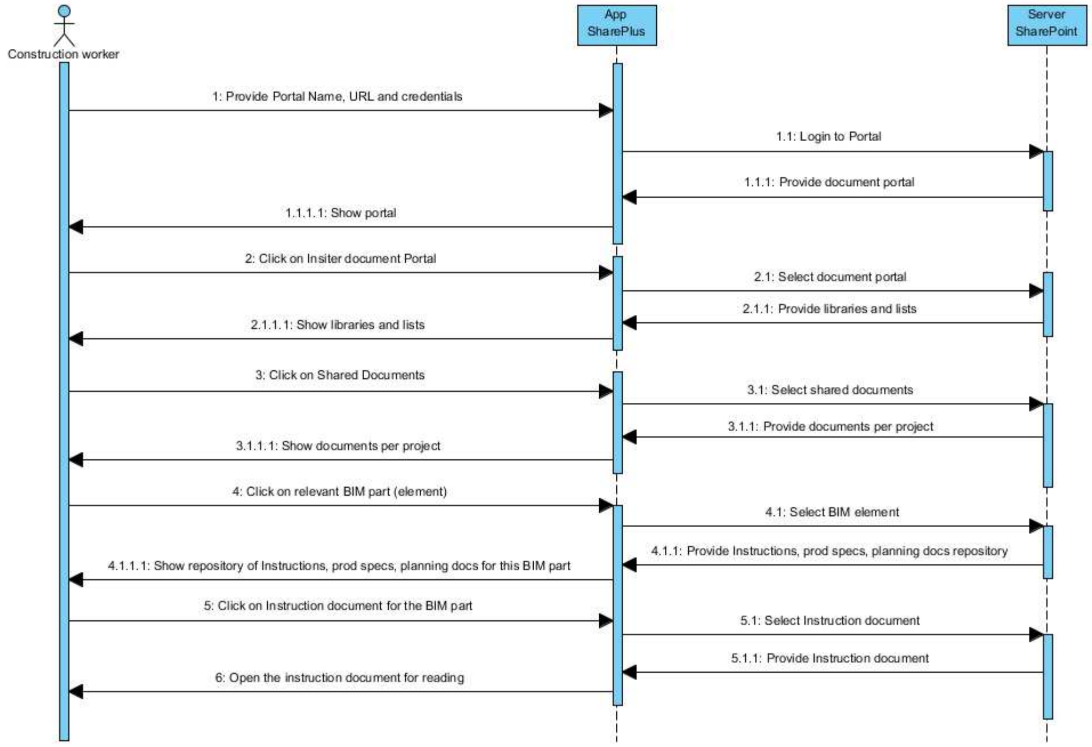

Construction worker
App SharePlus
Server SharePoint
1: Provide Portal Name, URL and credentials
1.1: Login to Portal
1.1.1: Provide document portal
1.1.1.1: Show portal
2: Click on Insiter document Portal
2.1: Select document portal
2.1.1: Provide libraries and lists
2.1.1.1: Show libraries and lists
3: Click on Shared Documents
3.1: Select shared documents
3.1.1: Provide documents per project
3.1.1.1: Show documents per project
4: Click on relevant BIM part (element)
4.1: Select BIM element
4.1.1: Provide Instructions, prod specs, planning docs repository
4.1.1.1: Show repository of Instructions, prod specs, planning docs for this BIM part
5: Click on Instruction document for the BIM part
5.1: Select Instruction document
5.1.1: Provide Instruction document
6: Open the instruction document for reading

Figure 4.4: UML sequence diagram between on-site worker, mobile app, SharePoint server software

4.2 Defect and quality issue registration tool

4.2.1 Functionality

An inspection tool will be developed for the registration of defects and quality issues. The tool will be provided with the next functionalities in order to fulfil the INSITER requirements:

1. Authentication

   The worker is able to log on to the system supplying his/her credentials

2. Navigation

   The worker is able to navigate to the part subjected to inspection via map, BIM viewer and/or part number

   The current location of the worker is shown on the map and the worker is able to select (one of) the building(s) around him/her for which a BIM model exists. The BIM model is visualised on the mobile device by means of an integrated BIM viewer. The worker navigates via the BIM viewer to the part subjected to inspection and clicks on that part. A panel becomes visible on which the worker is able to:

   - View the requirements in order to register the degree of compliancy, see also point 3
   - View additional information for this part, see also point 4

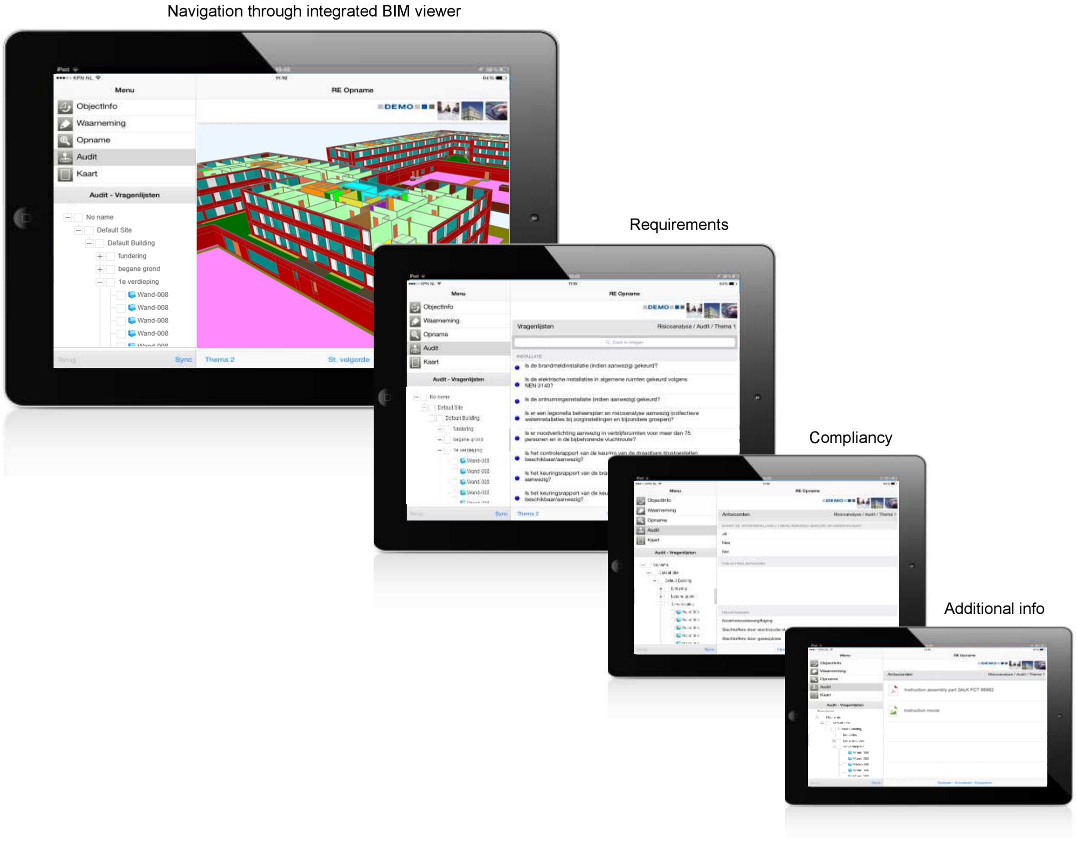


Figure 4.5: mobile screen examples

3. Presenting the requirements

   The requirements are shown for the selected part and the worker is able to register the compliancy to these requirements. The input provided by the user can be yes/no, a value, one or more predefined answers or an input string.

4. Show additional information

   The additional information for the selected part would be anything that is related to it. This can be documents, sub-models, measurements, pictures, sounds, instructions, IFC properties.

   Technically all these relationships have to be managed. For this purpose we introduce a linked data layer entity.

This toolset will be created by WP3 based on inputs coming from WP1 see section 8.3.

4.3 Augmented Reality tool

The application on the mobile device equipped with WIFI/4G will provide the following functionalities:

1. Access to dedicated AR/BIM server, where geometry of the BIM model will be downloaded or streamed to the client to be visualized on the client on basis of WebGL. The AR development concept includes a dedicated AR/BIM server design based on Fraunhofer Instant3DHUB technology to support server side rendering and web-based visualization.
2. Search for or navigate to the required (part of) the design model on the AR/BIM server.
3. Download or stream (part of) the design model.
4. Visualize (part of) the design model.
5. Click on (parts of) the model in order to retrieve additional information.
6. Superimpose the reality, as seen by the camera on the device, with relevant parts of the design model.
7. Adjustable transparency or viewing selection of the model (parts).
8. Click functionality on (parts of) the design model to be forwarded to external documents or web page that contains links to available documents or further links to other INSITER toolsets (e.g. the defect and quality issue tool).

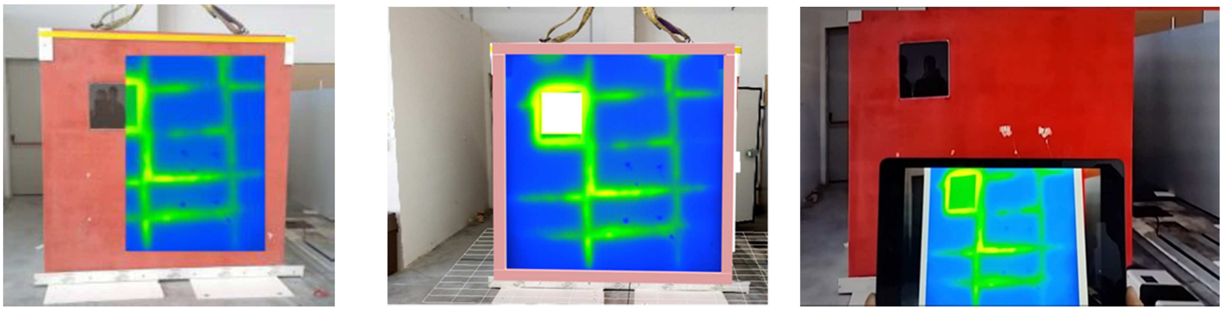


Figure 4.6: Augmented Reality toolset will being developed and tested at INSITER lab (WP2 and WP5)

4.4 Dashboard for KPIs

A dashboard is required to visualize the KPIs that are defined for INSITER.

The dashboard can be accessed via web browser on any device. The user logs in (with username and password) and the dashboard appears:

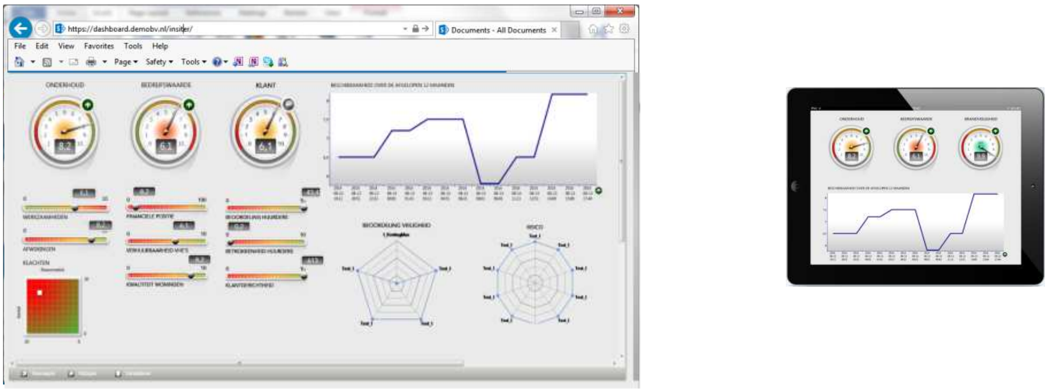

https://dashboard.demobv.nl/insiter/
Documents - All Documents
File Edit View Favorites Tools Help
Page Safety Tools
ONDERHOUD
BEDRIJFSWAARDE
KLANT

Figure 4.7: illustration of KPI dashboard

4.4.1 Setting up a dashboard

Data plays an important role in setting up a dashboard. Important is to determine which KPIs are to be visualized, which data is required, where the data is stored, how can it be accessed and what the refresh rate is (e.g. once a day/week/month/year, real time). The next step is to foresee a (automated) mechanism for acquiring the data with the desired frequency. Once that is established the dashboard software must be able to filter and aggregate the supplied data in order to provide a meaningful KPI value that can be displayed. The KPI value can be displayed in several ways for instance as a meter, as spider, trend graph, measure bar.

This dashboard will be configured by WP3 based on KPI inputs coming from WP1 see section 8.3.

Authorization to select/create/update and delete a dashboard will be provided to the project manager and/or the functional application manager. The rest of the project team members are only authorized to select a dashboard for viewing.

4.5 Import and conversion of measurements data

Each device has normally a raw data format resulting from a scan or reading by the hardware itself. This raw data format is very often already a standard itself but containing many implicit assumptions. This is why almost any vendor of hardware/measurement devices offers software for post processing itself. This software takes the raw data and device specific implicit assumptions to convert the data in something more relevant and useful for external applications. In the area of point clouds there is only one open standard, i.e. E57, however there are many alternative formats. In virtually any case it is beneficial to use the vendor supplied post processing software to enrich and convert the raw data. The resulting data can be sent to one of the databases, i.e. the INSITER BIM platform. Dedicated INSITER applications will then be able to connect the BIM model and the created data to show the results from the self-inspection in an integrated way.

More details about the formats can be found in deliverable D4.3.

4.6 Model checking

4.6.1 Software environment

The tool will be embedded in the 3D software desiteMD by Ceapoint. The software desiteMD offers the possibility to show an additional window on the user interface which may contain extra functionalities. These functionalities are created using web technologies– JavaScript, HTML and CSS. To interact with the main software, an API is provided that can be used from within the JavaScript code.

The model will be downloaded as an IFC file from the Open BIM server. Data can be read from and written to the dedicated INSITER database.

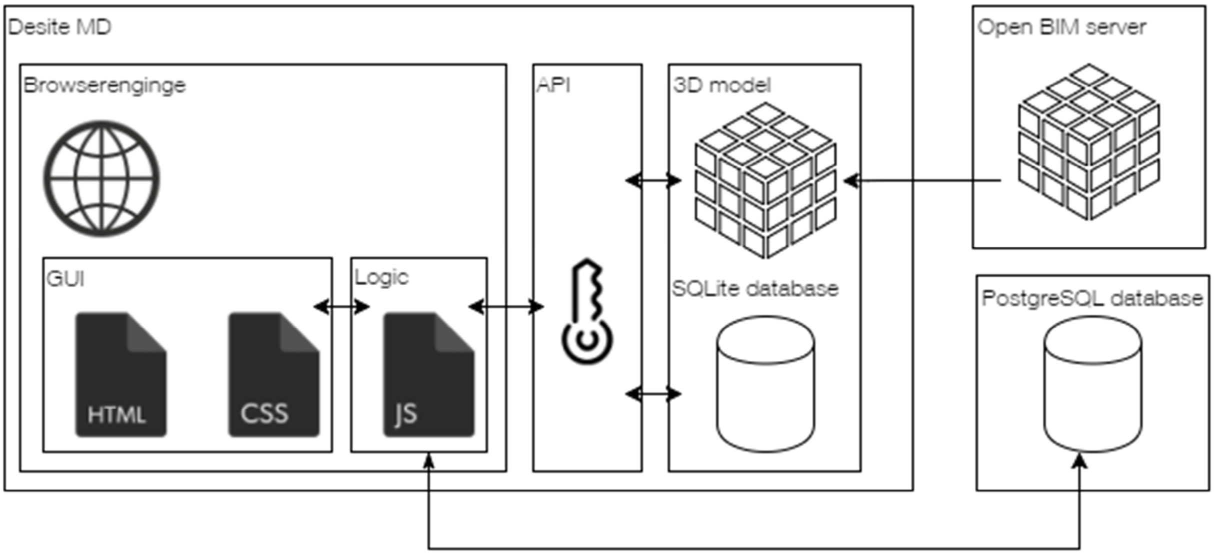


Figure 4.8: overview of software environment

4.6.2 Process

There are three different key players in the process:

1. The BIM expert, who needs to ensure the quality of the model and its data,
2. The modeller, who provides the model and
3. Other engineers or specialists, who deliver the data.

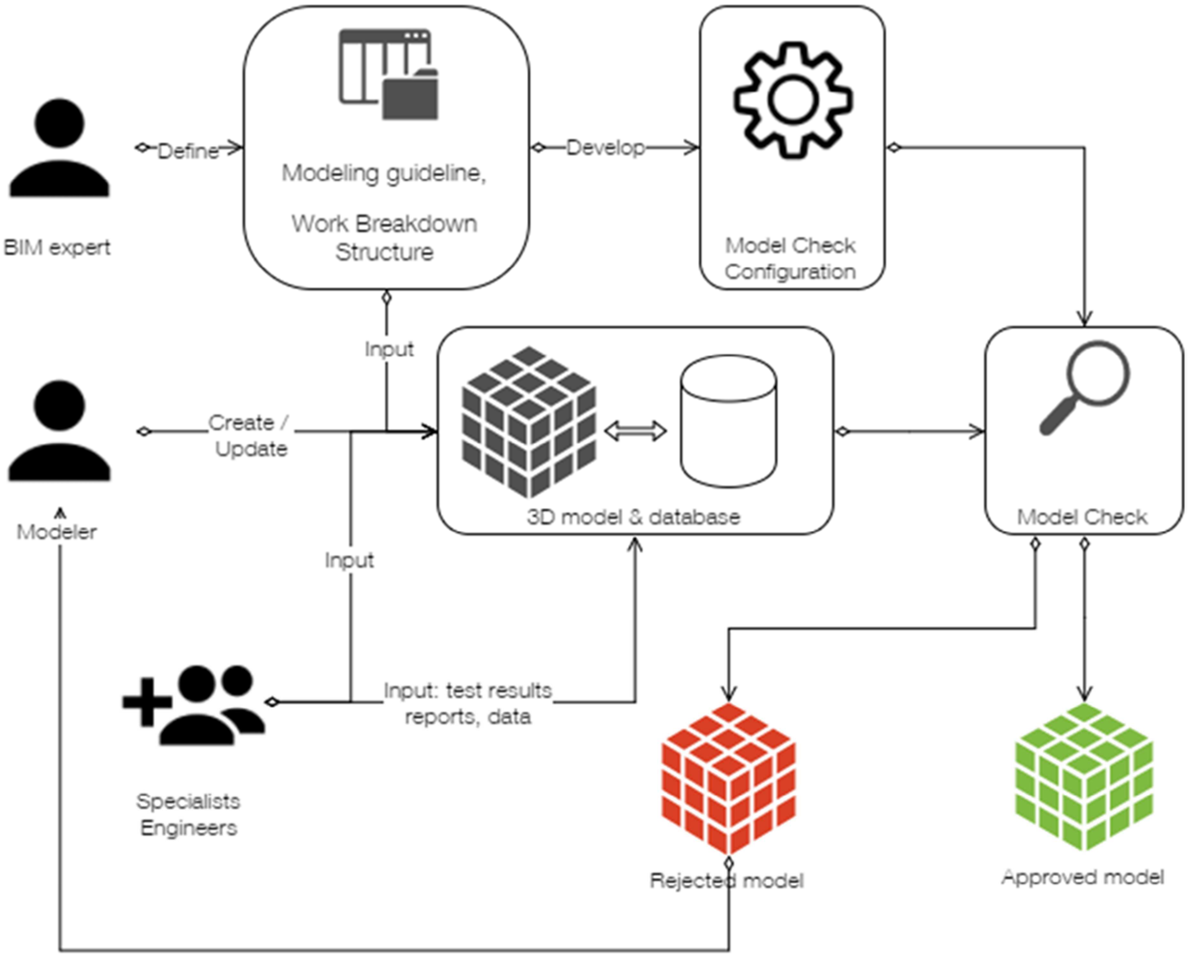


Figure 4.9: overview of main actors and roles

The tool is configured by the BIM Expert based on the modelling guideline. First tests are then run by the modeller before he or she makes the model available to the other project stakeholders. Specialists and engineers add data to the model directly or give the data to the modeller to integrate them. Afterwards the model will be checked again for final approval.

4.6.3 Main process

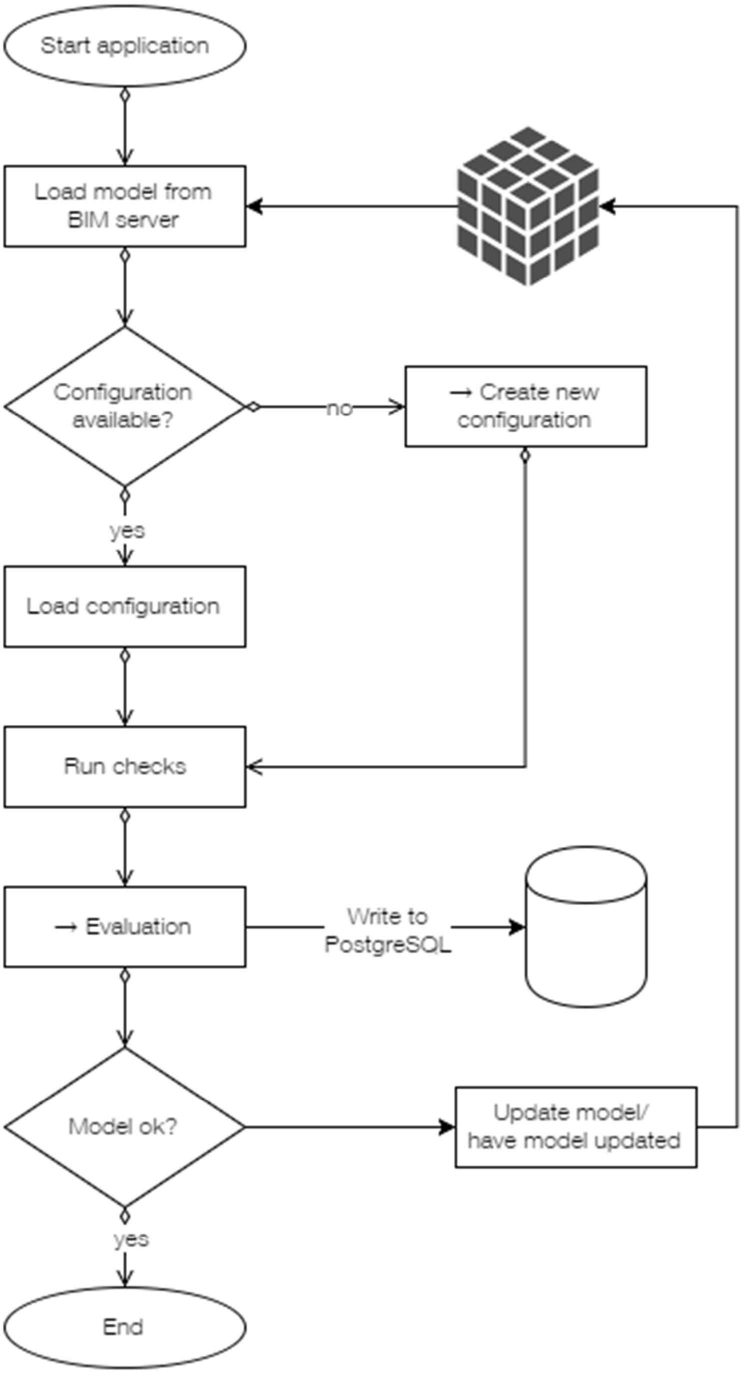


Figure 4.10: process flowchart

DesiteMD as the host application is started by the user who then loads the model from the BIM server. If a configuration for the Model Checker is available, the configuration is loaded into the tool. The user can select and run the pre-configured checks. The result will then be evaluated and based on these checks the model can be approved or rejected and handed back to the modeller for rework.

4.6.4 Create configuration

If a configuration does not yet exist, the model is analysed by the tool and the user can create checks. A check consists of at least one task. If one task fails for an object, the subsequent tasks will be skipped.

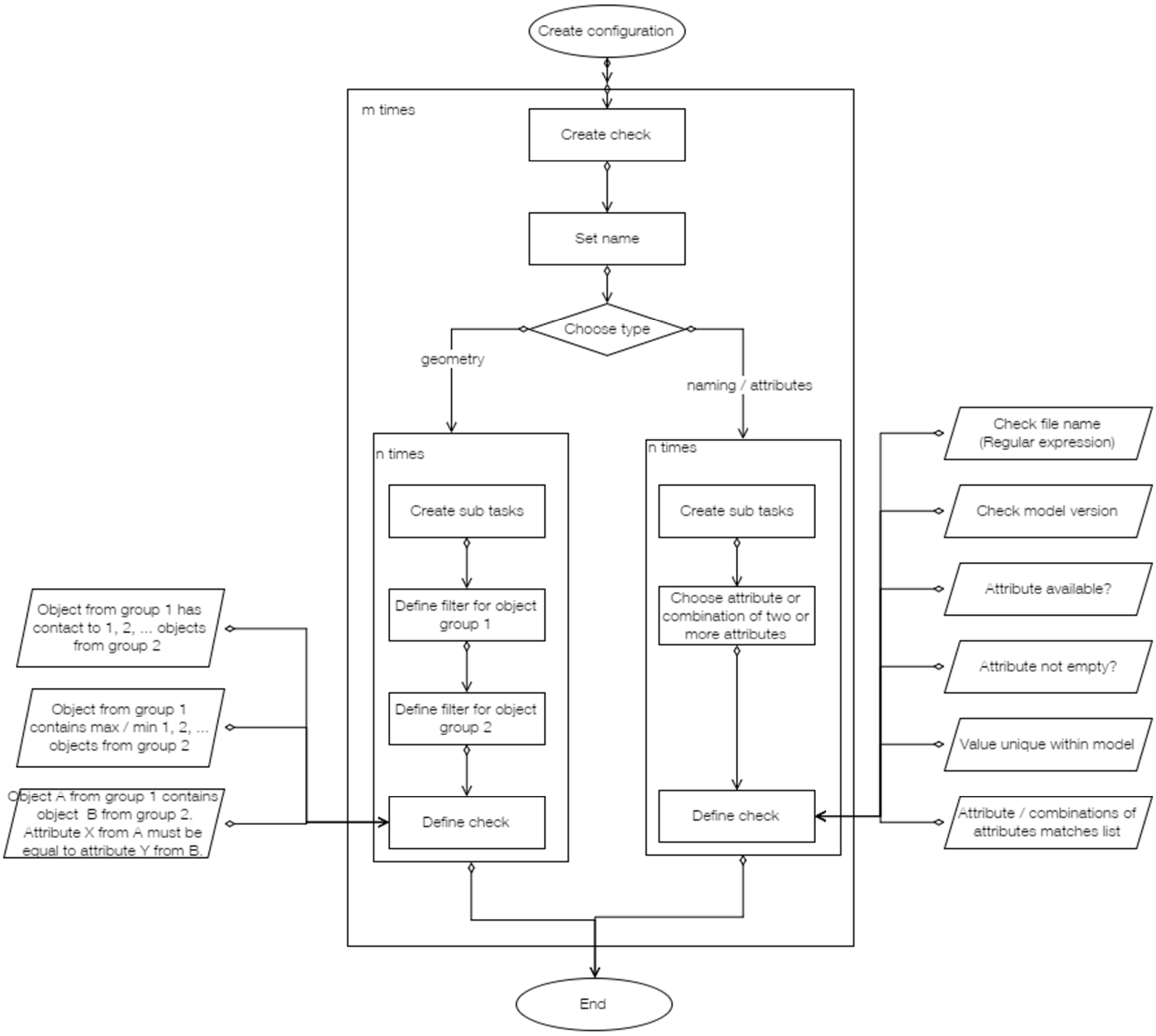
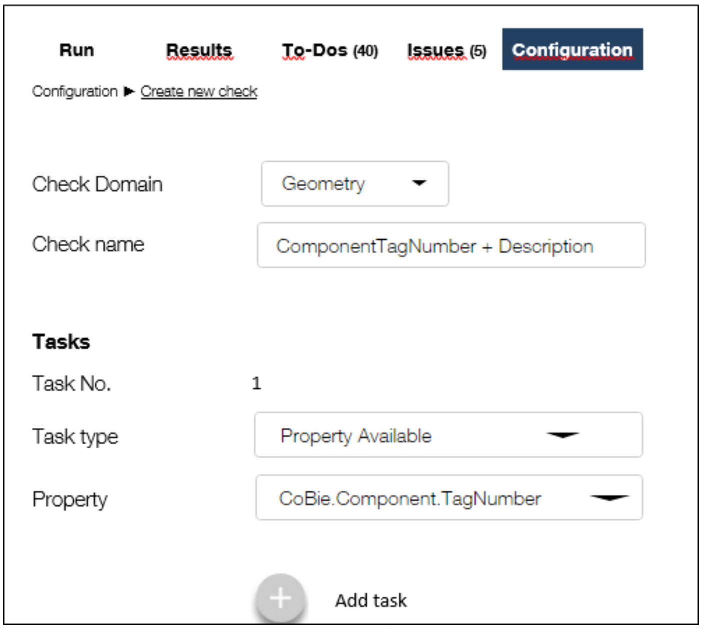

Create configuration
m times
Create check
Set name
Choose type
geometry
naming / attributes
n times
Create sub tasks
Define filter for object group 1
Define filter for object group 2
Define check
n times
Create sub tasks
Choose attribute or combination of two or more attributes
Define check
Object from group 1 has contact to 1, 2, ... objects from group 2
Object from group 1 contains max / min 1, 2, ... objects from group 2
Object A from group 1 contains object B from group 2. Attribute X from A must be equal to attribute Y from B.
Check file name (Regular expression)
Check model version
Attribute available?
Attribute not empty?
Value unique within model
Attribute / combinations of attributes matches list
End

Figure 4.11: configuration flowchart

The following diagram shows the general process of model checking:
For defining model checks and to apply it to BIM models it is aimed to develop a front-end for user-friendly setup of model checking rules. I possible interface may look as follows:



Check Domain: Geometry

Check name: ComponentTagNumber + Description

Tasks

Task No. 1

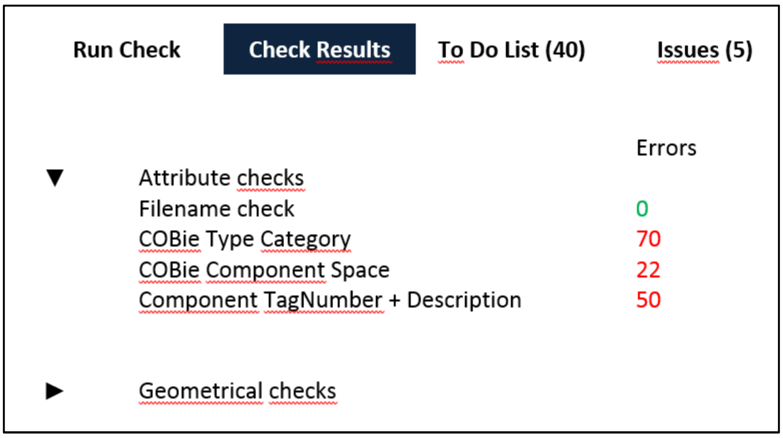
Task type: Property Available

Property: CoBie.Component.TagNumber

Figure 4.12: model checking interface

4.6.5 Evaluation of test results

The check results are shown in a list. Next to each check name, the number of failed elements is listed and the user can enable highlighting. Clicking on the check name, the user sees a detail view which objects failed. He or she can zoom to each individual item and add the issue to a virtual to-do-list or open a ticket for clarification

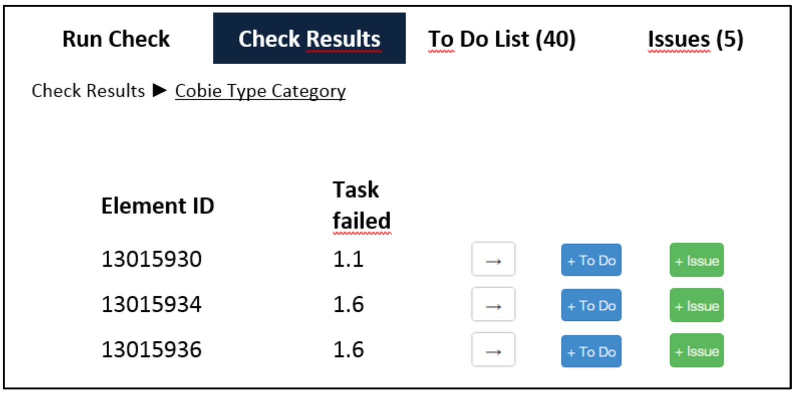


		Errors
▼	Attribute checks	
	Filename check	0
	COBie Type Category	70
	COBie Component Space	22
	Component TagNumber + Description	50
▶	Geometrical checks	

Figure 4.13: result checking interface

Also, the results can be exported into a file with a structure similar to this, for example an excel file:

```
ID          CheckName                                Status     FailedTask
   13015926 CoBieComponentTagNumber + Description     OK
   13015928 CoBieComponentTagNumber + Description     OK
   13015930 CoBieComponentTagNumber + Description     OK
   13015936 CoBieComponentTagNumber + Description     Failed     1.6
   13015939 CoBieComponentTagNumber + Description     Failed     1.6
```



Element ID	Task failed			
13015930	1.1	→	+ To Do	+ Issue
13015934	1.6	→	+ To Do	+ Issue
13015936	1.6	→	+ To Do	+ Issue

Figure 4.14: result checking interface IDs

4.6.6 Check Types

Model checks can be clustered into 3 kind of checks: global checks, attribute checks and geometrical checks.

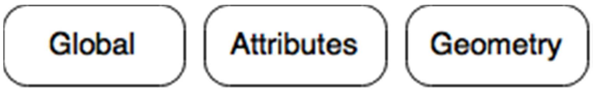


Figure 4.15: main clusters of model checking

Global checks cover checking (comparison with requirements) of:

- Naming of the model file itself
- Naming of Layers
- Naming of Families & Types
- Naming of Elements (e.g. doors, windows)
- Uniqueness of Element ID (1 ID for only 1 element in the model)
- Model versions: Comparison of 2 versions of the model: What has been changed/deleted/added?

Attribute checks are the most time saving investigation for the model checking process. Manually done, this takes several hours, or even days, to run through all kind of attributes. By using the QC tool, this is only a question of minutes. But to start these checks, the global checks need to be finalized, because the attribute checks take the uniqueness of the Element ID as granted, otherwise a distinct allocation of the 3D elements will not be possible. Provided, that this condition is fulfilled, the attribute checks give the possibility to examine the following:

- Existence of all required attributes in the model
- Assignment of necessary attributes to the correct elements

- Verification of attribute values with requirements
- Correctness of attribute combinations, mapped to 1 element
- Number of occurrence of attributes

  How often is an attribute set in the model? Is it only set once?

Geometry Checks also provide several opportunities:

- Investigation, if all elements in the model are closed 3D elements.
- That means, do they have a volume? This is very important for quantity take off, for example.
- Valid correlation of elements to a specific level, room or other elements
- Contact Check of elements

  e.g. Does the wall ends at a floor or is it modelled continuously through the floor?

Output of all this is a report in terms of lists and a coloured model to easily identify the issues.

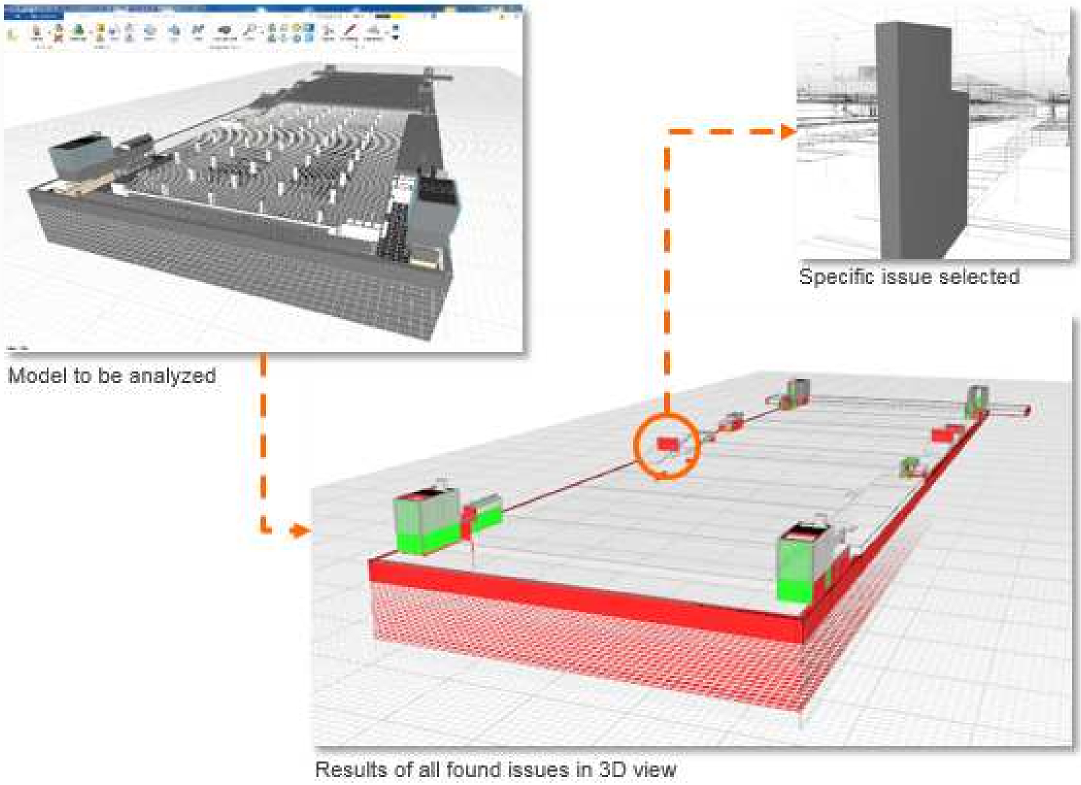
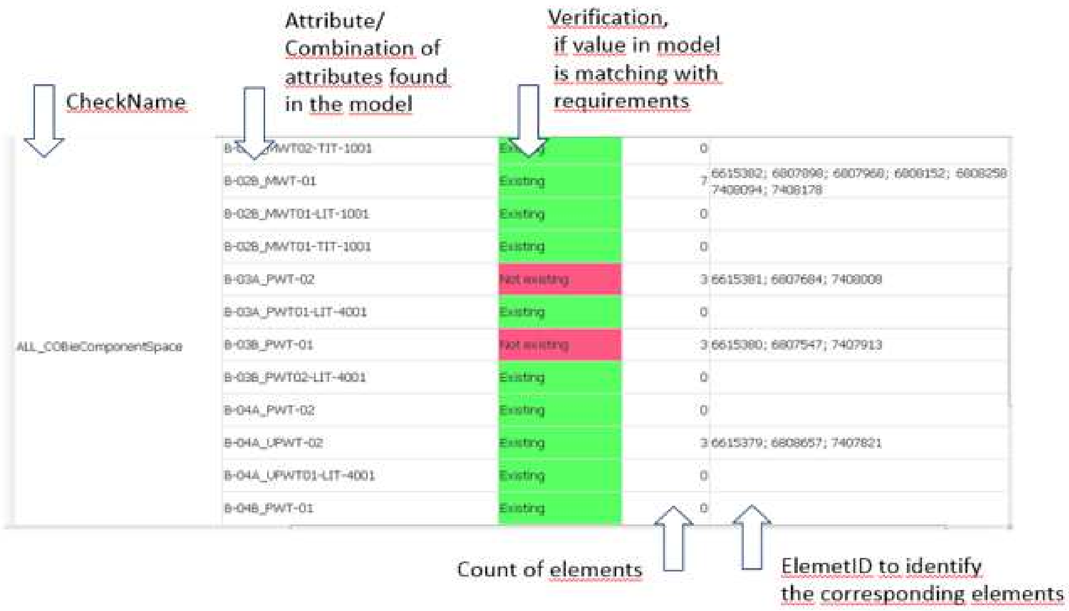


Figure 4.16: example of geometry checking



CheckName

Attribute/ Combination of attributes found in the model

Verification, if value in model is matching with requirements

ALL_COBieComponentSpace	B-02B_MWT02-TIT-1001	Existing	0	
	B-02B_MWT-01	Existing	7	6615382; 6807898; 6807968; 6808152; 6808258 7408094; 7408178
	B-02B_MWT01-LIT-1001	Existing	0	
	B-02B_MWT01-TIT-1001	Existing	0	
	B-03A_PWT-02	Not existing	3	6615381; 6807684; 7408008
	B-03A_PWT01-LIT-4001	Existing	0	
	B-03B_PWT-01	Not existing	3	6615380; 6807547; 7407913
	B-03B_PWT02-LIT-4001	Existing	0	
	B-04A_PWT-02	Existing	0	
	B-04A_UPWT-02	Existing	3	6615379; 6808657; 7407821
	B-04A_UPWT01-LIT-4001	Existing	0	
	B-04B_PWT-01	Existing	0	

Count of elements

ElemetID to identify the corresponding elements

Figure 4.17: example of model checking list

4.6.7 System environment

The tool can run on both local Windows machines and a Windows Terminal Server Environment. The recommended way of use is the Terminal Server environment. It's a controlled environment, with good performance and shared resources.

Running the tool locally is still possible, but lacks user management and a common data store.

Working locally the workflow will look as follows:

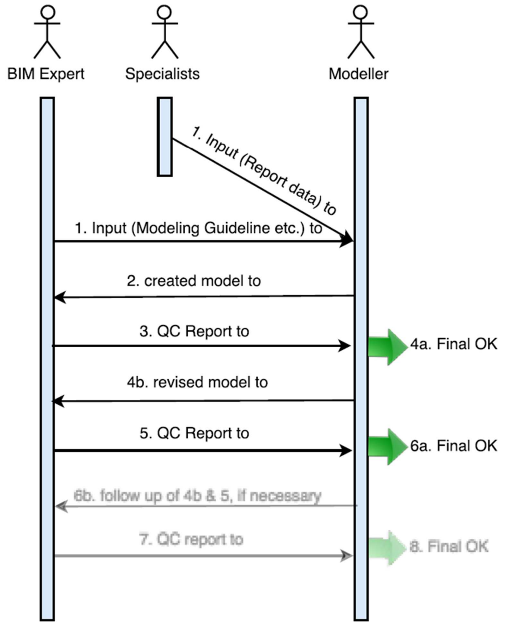


Figure 4.18: system environment for model checking

Working with windows terminal server environment, the workflow will look as follows:

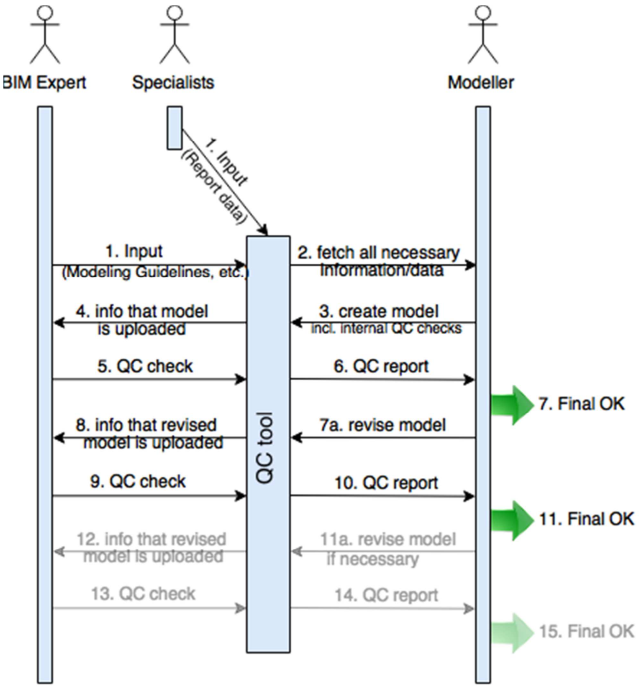

BIM Expert
Specialists
Modeller
1. Input (Report data)
1. Input (Modeling Guidelines, etc.)
2. fetch all necessary information/data
3. create model incl. internal QC checks
4. info that model is uploaded
5. QC check
6. QC report
7. Final OK
7a. revise model
8. info that revised model is uploaded
QC tool
9. QC check
10. QC report
11. Final OK
12. info that revised model is uploaded
11a. revise model if necessary
13. QC check
14. QC report
15. Final OK

Figure 4.19: workflow within the system environment

4.7 Clash detection

4.7.1 Software environment

The general process of the clash detection is related to three tools:

- Navisworks Manage: The core for analysing 3D models regarding geometrical conflicts.
- 3D DCA (design coordination analyser) is a Navisworks plugin which eases the way of doing a clash detection and prepares the results for the design coordination system.
- 3D DCS (design coordination system) is a web based platform. This platform is an information system that provides information about identified design conflicts of a clash detection analysis and offers the possibility to coordinate them.

4.7.2 Process

The following scheme gives an overview of the high level process. In general there are three major roles connected to the clash detection:

- The Designer(s) will provide the models and are responsible for the correctness of the data
- The Clash Detection Expert will coordinate the process of the clash detection and perform the clash analysis
- The Design Coordinator will coordinate the identified issues by the clash expert and initiate counter measures

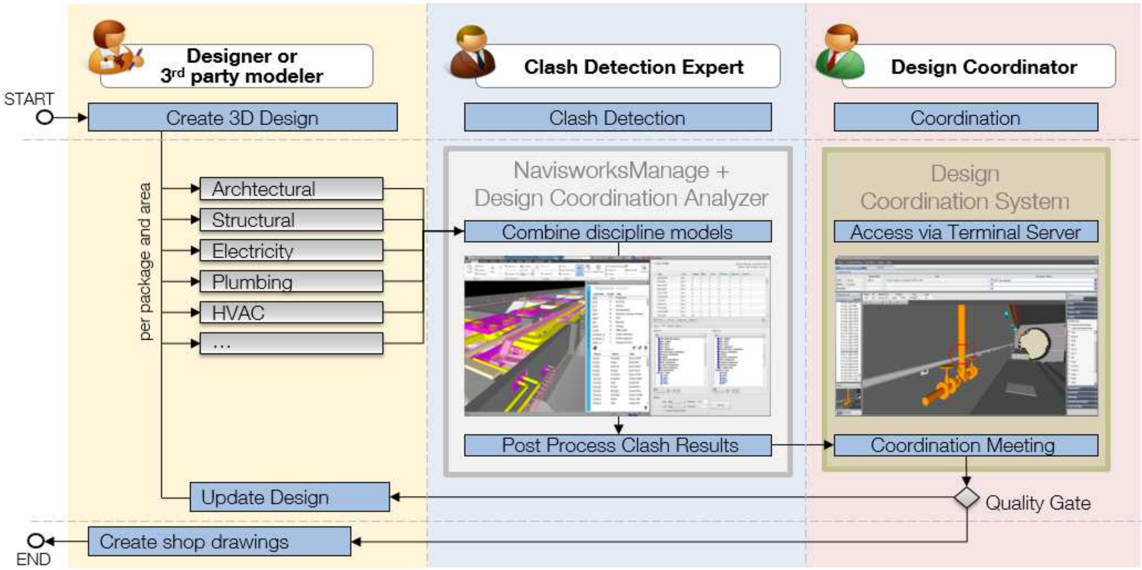


Figure 4.20: process overview for BIM clash detection

The following diagram shows the sequences of the clash detection process in are more detailed way and points out the interactions between the involved personnel and the used tools.

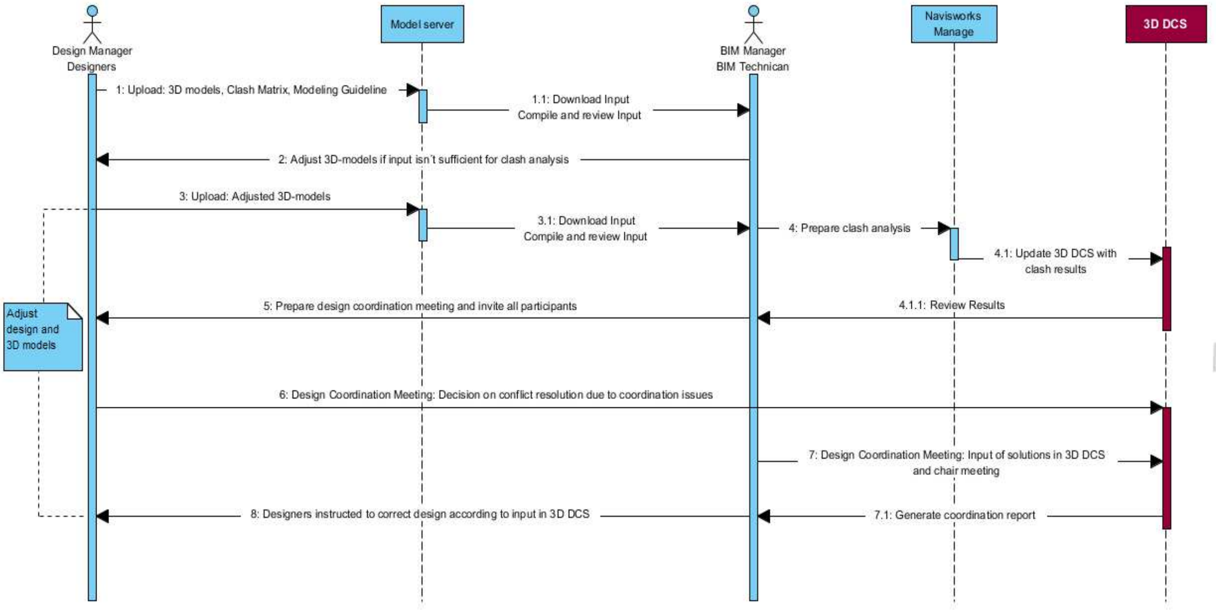

Design Manager
Designers
Model server
BIM Manager
BIM Technican
Navisworks
Manage
3D DCS
1: Upload: 3D models, Clash Matrix, Modeling Guideline
1.1: Download Input
Compile and review Input
2: Adjust 3D-models if input isn´t sufficient for clash analysis
3: Upload: Adjusted 3D-models
3.1: Download Input
Compile and review Input
4: Prepare clash analysis
4.1: Update 3D DCS with
clash results
5: Prepare design coordination meeting and invite all participants
4.1.1: Review Results
Adjust
design and
3D models
6: Design Coordination Meeting: Decision on conflict resolution due to coordination issues
7: Design Coordination Meeting: Input of solutions in 3D DCS
and chair meeting
8: Designers instructed to correct design according to input in 3D DCS
7.1: Generate coordination report

Figure 4.21: overview of clash detection process

4.7.3 Data Exchange

One of the most challenging issues of the clash detection process is the data exchange. While Navisworks offers the possibility to import many different file formats it is very easy to initiate the clash detection process. More challenging is the process of coordinating the identified issues and to forward the results to the corresponding responsible persons. The following diagram shows the already available way of exchanging data and in addition to this the interfaces which have to be developed to attach the system to the INSITER toolset. The orange coloured arrows show the exchange interfaces which have to be developed/enhanced:

- Straight arrow: Shows the interaction between the BIM Model Server and Navisworks for initiating the Clash Detection process. This interaction is already available in a manual way. It has to be investigated if there is a way of automating this exchange.
- Interrupted arrow: Shows the forwarding of coordination data to the INSITER databases to provide the site workers with instructions to be applied on site.

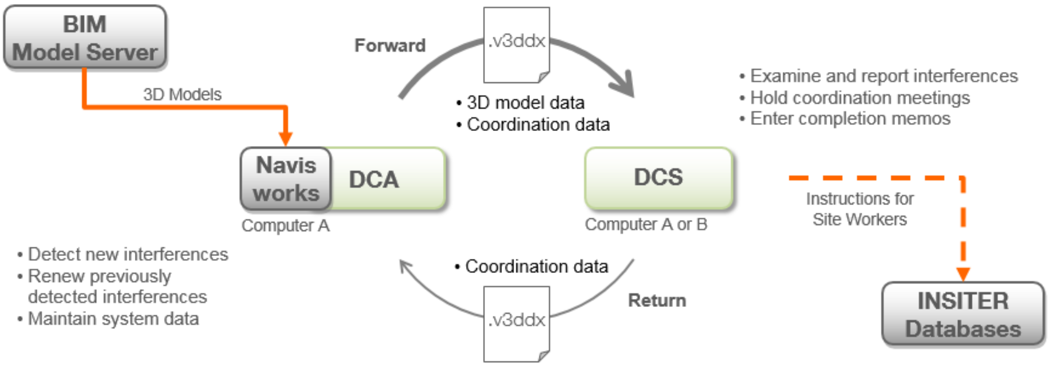


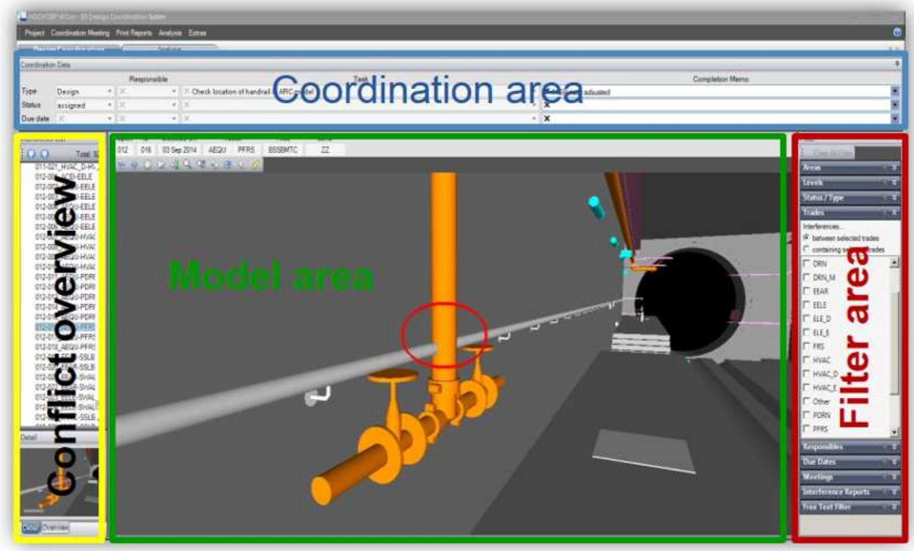
Figure 4.22: overview for data exchange

4.7.4 User Interface of the 3D DCS (Design Coordination System)

To identify the relevant information which has to be transferred to the INSITER database it is essential to know about the information which will be entered into the Coordination system and which will be added to the identified conflicts. The following section gives an overview of the coordination data which will be gathered in a regular coordination meeting.

The user-interface of the 3D DCS is split into four areas. These areas are:

- Model area – in this area it is possible to move the 3D model to look at the conflict in more detail
- Filter area – here it is possible to filter according to various criteria (e.g. trade, status, report number, responsible)
- Conflict overview – in this area there are all information about the conflicts
- Coordination area – in this area all coordination information will be entered during the coordination meeting

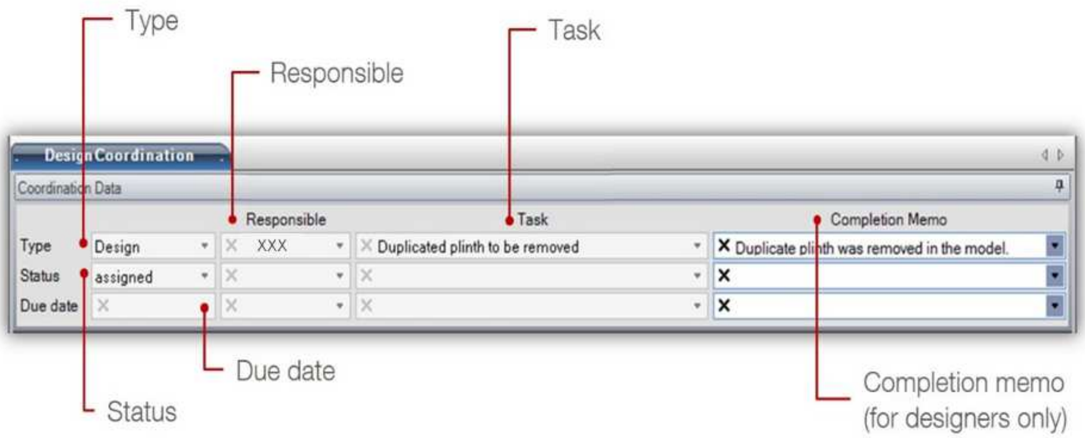


Type Design
Status assigned
Due date
Check location of handrail ARC model
adjusted

Figure 4.23: example of user Interface of the 3D DCS (Design Coordination System)

All coordination information will be shown in the “coordination data”. During the design coordination meeting the fields will be filled with information. These information are:

- Type: Drop down menu to choose the kind of clash
- Status: Drop down menu to choose the status of the clash
- Responsible: Drop down menu to choose the responsible stakeholder for the resolution of the clash
- Task: Text field to enter countermeasure to solve the issue
- Completion Memo: Text field responsible persons to inform the team about performed actions
- Due date: Selection field (calendar) for due date of the coordinated task to be performed



		Responsible	Task	Completion Memo
Type	Design	XXX	Duplicated plinth to be removed	Duplicate plinth was removed in the model.
Status	assigned			
Due date				

Figure 4.24: main fields of coordination data

In the field type there are five selections that can be chosen. These selections are:

- Undefined: The default status of each new clash before it is reviewed or no other type defined
- Design: The problem is a design issue that can be resolved by the design team involved
- Modelling: The clash is a graphical issue that can be resolved by the modelling team
- On Site: The situation must be taken care of on site
- Ignorable: The clash is insignificant and negligible

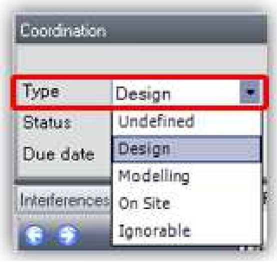
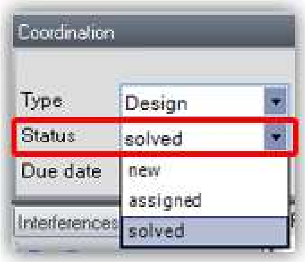


Figure 4.25: menu for type selection

For the status of the clash three selections are possible:

- New: The clash is a newly identified clash that has not been reviewed or assigned before
- Assigned: The clash requires actions and is assigned to a responsible person
- Solved: The clash is solved and removed from list and will not be tracked further



Type: Design
Status: solved
Due date

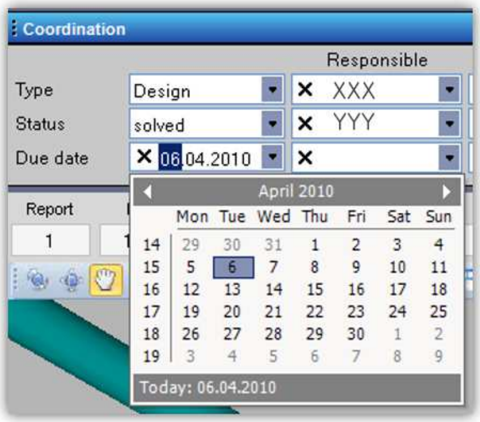
Figure 4.26: menu for clash status selection

In the field “Due date” the selection of date for deliverables to be submitted to BIM Team for clash analysis update. A pull-down monthly calendar is displayed along with the week number on the left side column.

In the selection field “Responsible” there is a list with all involved stakeholders. During the coordination meeting there can be selected which stakeholder should take care on the resolution of the clash.

The text field “Task” provide the location where a brief description of the tasks and actions that the responsible party(s) will carry in order to coordinate or solve a clash. The order of the tasks should follow the order of the responsible parties.

In the field “Completion Memo” provides a location for 3D DCS users (designers) responsible for the current clash to log-on and provide a brief comment / note of the situation allowing the other parties to be informed of the current actions that are taking place.



		Responsible
Type	Design	XXX
Status	solved	YYY
Due date	06.04.2010	

April 2010

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
14	29	30	31	1	2	3	4
15	5	6	7	8	9	10	11
16	12	13	14	15	16	17	18
17	19	20	21	22	23	24	25
18	26	27	28	29	30	1	2
19	3	4	5	6	7	8	9

Today: 06.04.2010

Figure 4.27: menu for time planning selection

4.7.5 Reporting

3D DCS allows the user to create different types of reports, depending on the user level. The different report types are:

- Executive Report — gives an overview on the project progress (without showing any identified issues) This kind of report shows an overview of the status of 3D coordination of a project. The report can be created by anyone except designers. These overview reports are particular for the upper management and project manager to inform about the status of the 3D coordination of the whole project.
- Coordination meeting summary — provides information about all coordinated issues of a specific coordination meeting.

Figure 4.28: structure of 3D DCS report

The coordination meeting summary reports are particular for the project management and design coordinators. This kind of report provides a detailed view of all interferences reviewed in one coordination meeting. Report can be created at any time by anyone except designers.

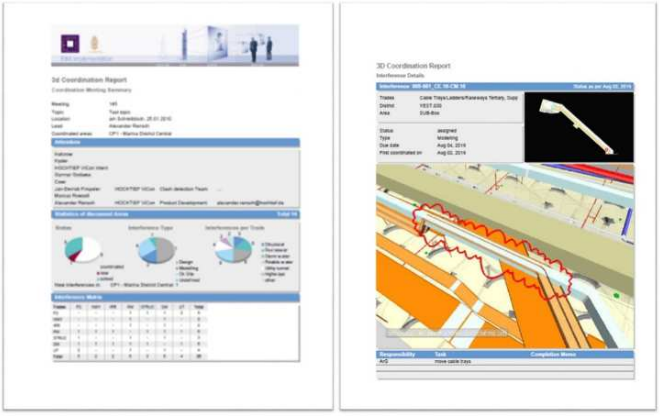


Figure 4.29: example of 3D DCS summary report

- Individual selection report gives the possibility to report issues of selected issues (via filter options)

  The individual selection report can be used by any user of the 3D DCS. This kind of report provides a detailed view of filtered interferences reviewed in the coordination meeting. So if a user for example only wants to have a report where all conflicts are shown which belong to the trade electrical, he can filter according to this trade and create a report.

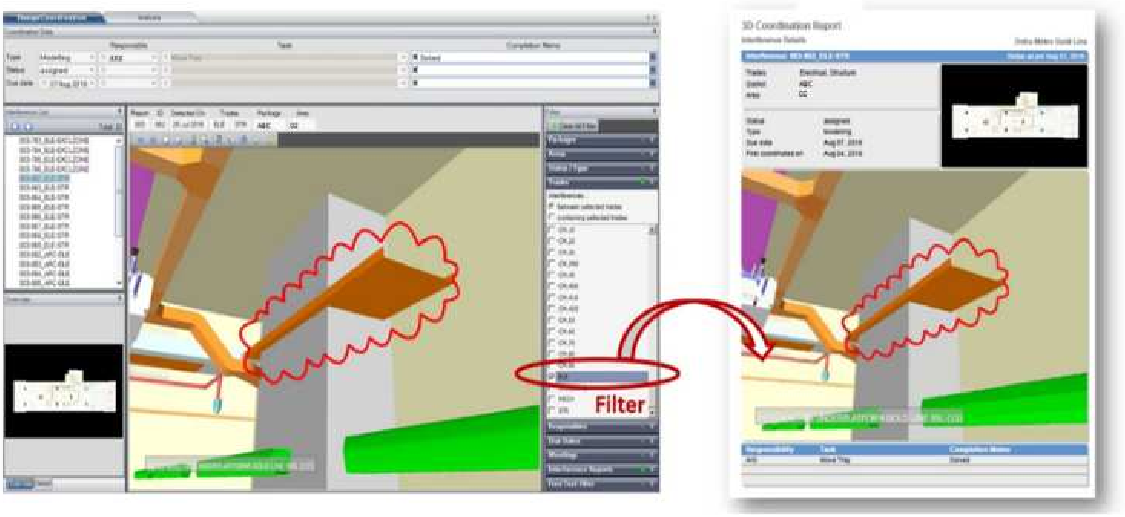


Figure 4.30: example of detailed clash detection report

5. Comparative analysis with state-of-the-art software

The INSITER toolset will utilize existing software for supporting the INSITER methodology. For this purpose, relevant state-of-the-art software candidate will be reviewed; commercial as well as partners' software. The decision which software will be used is based on the scope of the project, available knowledge and experience, interoperability with BIM and other software packages.

5.1 File Sharing System software

Tools Characteristics	Bitrix24	Collaboration/File sharing	SharePoint
Company	Bitrix24	Huddle	Microsoft
Purpose of the product	for social collaboration, communication and management tools for organizations	A software solution for managing files, tasks and team communication	A software solution for document management, websites, collaborative workspaces
Benefits	Cloud or self-hosted solution Access to source code Also for mobile devices	Cloud solution Also for mobile devices	Cloud or self-hosted solution Open standard CMIS interface Also for mobile devices Knowledge available because SharePoint is already in use by several INSITER partners
Relevancy to INSITER	A software solution as follows is required for the storage of inspection and instruction documents and templates	A software solution as follows is required for the storage of inspection and instruction documents and templates	A software solution as follows is required for the storage of inspection and instruction documents and templates
Challenges	It is not able to interface via the CMIS standard protocol. As there is access to the source code this open standard could be added to the software	It is not able to interface via the CMIS standard protocol. As there is no access to the source code this open	This software can be used as is

Interoperability issues	CMIS standard interface should be added to the software	No CMIS interface can be added to the software	CMIS standard interface is supported therefore no issue
Reference	https://www.bitrix24.com/alternatives/free-sharepoint-alternative.php?gclid=CLXDk_rUrMwCFQeVGwodDpsK_Q	https://www.huddle.com/	https://products.office.com/en-GB/sharepoint?legRedir=true&CorrelationId=461ba077-b342-4c98-8c6d-e2f995c6a145

INSITER conclusions and decision for software development:

The chosen solution is the already available SharePoint software with open standard CMIS which will be configured to support the functionalities as described under section 4.1

SharePoint is a web application platform in the Microsoft Office server suite with functionalities for intranet, extranet, content management, document management, workflow management and web content management.

5.2 Augmented reality software

For Augmented reality software there are different products on the market available like Vuforia, Wikitude and Instant3Dhub/Visionlib. The table below shows the characteristics of each software package.

Tools Characteristics	Vuforia	Wikitude SDK	Instant3Dhub/Visionlib
Company	PTC	Wikitude	Fraunhofer
Purpose of the product	Augmented Reality (AR) Software Development Kit (SDK)	AR Software Development Kit (SDK) focusing on image recognition and tracking	Instant3DHUB: Server side rendering service with a pre-processing of the INSITER BIM models by Instant3DHUB on the server side. The relevant geometry of the BIM model will be streamed to the client to be visualised on the client on basis of WebGL. VisionLIB technology: Computer vision based tracking solution with a combination of 3D model object and marker-based tracking methods.
Benefits	Supports a variety of 2D and 3D target types including markerless image targets. Developers can build Vuforia apps in Eclipse, Xcode and Unity.	Easy to develop basic and simple AR applications with Wikitude Studio. Developers can integrate Wikitude's computer vision engine into an application based on Unity. Geo-referenced data supported.	Real-time visualisation of massive data sets and complex BIM models, possibility to combine multiple tracking methods and sophisticated tracking technology. Solutions can be provided by Fraunhofer as project partner and not by external software providers

Relevancy to INSITER	Robust 2D image based tracking as well as 3D object based tracking for small objects are supported. Vuforia applications support a variety of devices such as Android and iOS smartphones and tablets as well as digital eyewear. Vuforia could be tested concerning its usability and development capabilities for INSITER AR developments.	Mainly 2D image based tracking solutions are supported until now. First 3D object-based tracking solution is available as beta version for small objects only. Thus, Wikitude is currently not in the scope of the INSITER project.	The INSITER AR prototype can be built on a web-components based framework and standards such as X3DOM as well as a visual computing as a service infrastructure to deliver an interactive 3D data visualization solution. Instant3DHUB utilizes an adaptive combination of client, server and hybrid visualization techniques, while orchestrating transmission, caching and rendering services in order to deliver complex data sets on any device class and network architecture. Besides, VisionLIB supports an advanced and combined 2D image and 3D model based tracking technology. Moreover, a variety of devices could be supported, as the application to be developed will be web-based. Thus, this Fraunhofer solutions will enable the development of a new class interactive AR application prototype for INSITER.
Challenges	No IFC files are supported, rendering and tracking capabilities are limited.	No IFC files are supported, rendering and tracking capabilities are limited.	Further Development of Instant3DHUB services, functionalities and user interface as well as adaption and integration of the VisionLIB tracking technology. Data access, data integration and referencing of INSITER instrumentation device data (e.g.: thermal images) and other data for self-inspection and self-instruction to BIM/IFC objects

Interoperability issues	Import of files for most common 3D formats: OBJ, .FBX, .3DS, COLLADA files, no IFC support.	Import of files for most common 3D formats: OBJ, .FBX, .3DS, COLLADA files, no IFC support.	Import of files for 3D: X3D and IFC are supported to be imported. Export of data is not foreseen. During INSITER AR prototype development, the following formats are planned to be supported additionally: • Images (Format png) • Point clouds (Format e57) • 3D-models and Animations (X3D) • IFC model parameters, if applicable. The AR software prototype will also enable the access of other external planning documents (e.g. PDF) on file share systems (e.g. SharePoint) linked within the BIM model (part/object) as references (URL with GUID).
Reference	http://www.vuforia.com/	http://www.wikitude.com/developer	http://instant3dhub.org/ , https://www.igd.fraunhofer.de/Presse/Presseinformationen/Fraunhofer-ueberzeugt-mit-Augmented-Reality-Technologien

INSITER conclusions and decision for software development:

The chosen solution is the Fraunhofer software framework based on Instant3DHUB technology and Fraunhofer VisionLIB AR software technology for advanced feature and object tracking. New AR solutions for INSITER will be developed and integrated within WP2. The INSITER AR prototype will be based on Fraunhofer technology and will be enhanced with the functionalities for AR as described in section 4.3., among others described in WP2. Moreover, additional AR Software Development Kits (SDK) such as Vuforia will be utilised and tested for INSITER as well. Besides live demonstrations, the to be developed INSITER BIM-based simulation and visualisation concepts as well as AR solution will be further described and presented within the deliverable 2.1 (M24) and deliverable 2.2 (M42).

5.3 Condition assessment software

Tools Characteristics	Ultimo	Ibis Main	RE Maintenance
Company	Ultimo	Ibis	DEMO Consultants
Purpose of the product	Software for condition assessments acc. to the NEN2667 Dutch standard	Software package that supports real estate managers with the inventory of maintenance needs	Software for condition assessments acc. to the NEN2667 Dutch standard
Benefits	Conformance with NEN 2767 standard Cost calculation	Conformance with NEN 2767 standard Cost calculation	Conformance with NEN 2767 standard Risk mitigation Deficiency prediction Cost calculation Software is adaptable to INSITER needs
Relevancy to INSITER	For renovation projects	For renovation projects	For renovation projects
Challenges	Communication with the BIM server for fetching construction parts is not available	Communication with the BIM server for fetching construction parts is not available	Communication with the BIM server for fetching construction parts is not available but can be foreseen as it is a product of DMO
Interoperability issues	The condition assessment software interface with the BIM server for importing building elements via the BIM server API has to be created	The condition assessment software interface with the BIM server for importing building elements via the BIM server API has to be created	The condition assessment software interface with the BIM server for importing building elements via the BIM server API has to be created. This is feasible as it is a product of DMO
Reference	http://www.ultimo.com/nl/software/totaaloverzicht-modules/conditiemetingen	http://www.ibis.nl/webshop/productoverzicht/cursus/conditiemeting-de-basis-detail	http://www.demobv.nl/downloads/Leaflet_RE_Maintenance_UK.pdf

INSITER conclusions and decision for software development:

The chosen solution is the Condition assessment RE Suite software (RE Maintenance) because it is multi language, configurable, already available within WP3 (DMO) and enhancements can be made to the software

5.4 Dashboard software

Tools	Dundas BI	Sisense	RE Dashboard
Characteristics			
Company	Dundas	Sisense	DEMO Consultants
Purpose of the product	Software package to create dashboards	Software package to create dashboards	Software package to create dashboards
Benefits	Works with multiple data sources For multiple devices supporting HTML5	Works with multiple data sources For multiple devices supporting HTML5	Works with multiple data sources For multiple devices supporting HTML5 Software is adaptable to INSITER needs
Relevancy to INSITER	Dashboard software is required to present and track the INSITERs KPIs	Dashboard software is required to present and track the INSITERs KPIs	Dashboard software is required to present and track the INSITERs KPIs
Challenges	Configuration of the data sources and KPI calculations	Configuration of the data sources and KPI calculations	Configuration of the data sources and KPI calculations
Interoperability issues	The data source of the dashboard needs to be connected to the BIM server and Mongo DB server	The data source of the dashboard needs to be connected to the BIM server and Mongo DB server	The data source of the dashboard needs to be connected to the BIM server and Mongo DB server
Reference	https://www.dundas.com/dundas-bi/	www.sisense.com/Dashboard-Software	http://www.board.com/en/capabilities-dashboarding

INSITER conclusions and decision for software development:

The chosen solution is the Dashboard RE Suite software (RE Dashboard) because it is flexible, configurable, multi-language and already available within WP3 (DMO) and enhancements can be made to the software.

5.5 Model checking software

Tools Characteristics	Board review essentials	IFC model checker	Desite MD
Company	BIM-review	Solibri (Nemetschek)	Ceapoint
Purpose of the product	To check the 3D model against design rules	To check the 3D model against design rules	To check the 3D model against design rules
Benefits	-	-	Already in use by HVC. Provides a powerful API with lots of possibilities to enhance the system according to given needs.
Relevancy to INSITER	Prevent model errors on site	Prevent model errors on site	Prevent model errors on site
Challenges	Not applicable	Not applicable	Not applicable
Interoperability issues	Not applicable, reads IFC files	Not applicable, reads IFC files	Not applicable, reads IFC files
Reference	http://www.bim-review.com/essentials.php	https://www.solibri.com/products/solibri-model-checker/	http://www.ceapoint.com/index.php?option=com_content&view=article&id=61&Itemid=55

INSITER conclusions and decision for software development:

The chosen solution is the Desite MD for model checking because it is already available within WP4 (HVC) and also knowledge on how to apply this toolset in an efficient way is available within the consortium.

5.6 Clash detection software

Tools Characteristics	3D clash detection	IFC Modelchecker	Navisworks
Company	Vico Software	Solibri (Nemetschek)	Autodesk
Purpose of the product	To check the 3D model on geometrical design conflicts	To check the 3D model on geometrical design conflicts (rule-based only)	To check the 3D model on geometrical design conflicts
Benefits	-	-	Already in use by HVC
Relevancy to INSITER	Prevent model errors on site	Prevent model errors on site	Prevent model errors on site
Challenges	Not applicable	Not applicable	Not applicable
Interoperability issues	Not applicable, reads IFC files	Not applicable, reads IFC files	Not applicable, reads IFC files
Reference	http://www.vicosoftware.com/products/3d-bim-software-clash-detection/tabid/229117/Default.aspx	https://www.solibri.com/products/solibri-model-checker/	http://www.autodesk.com/products/navisworks/overview

INSITER conclusions and decision for software development:

The chosen solution is the Navisworks enhanced with a plug-in by HVC for clash detection (D4.3) checking, because it is already available within WP4 (HVC).

5.7 Energy performance software

Monitoring software and whole building simulation software are used to assess the energy efficiency and thermal comfort performance of buildings. Building monitoring provides insight into the energy consumptions and HVAC installation in existing buildings and makes it possible to diagnose failures in an early stadium. Whole building simulation models can predict the performance of buildings, per zone, before or during the realization process. Additionally, these models take into account the impact of building adjustments on the performance. INSITER uses monitoring and simulation software to evaluate the impact of deviations, which are registered during self-inspection, in the realization process. To export readings from the INSITER software to the energy performance software, inoperability with BIM is an important requirement. The most optimal energy performance software for INSITER should be able to communicate bi-directional with the BIM model. In this way, monitoring and simulation results can be exported to the INSITER software and be registered in the defect analysis tool.

Based on the experience with energy performance software within the consortium, a selection of possible software packages was made that is suitable for integration with the INSITER software: VABI Elements, TRNSYS, Simaxx and RIUSKA. This section provides an overview of the purposes, benefits, relevancy, challenges and possible interoperability issues of these packages.

Tools Characteristics	VABI Elements	TRNSYS	Monavisa	RIUSKA
Company	VABI	TRNSYS	Simaxx	Granlund
Purpose of the product	Whole building simulation of the thermal comfort performance and energy performance of buildings	Simulation of thermal comfort performance and energy performance; performance of HVAC components;	Monitoring of Building and HVAC performance	RIUSKA is a software package for energy calculation and façade (building envelope) optimization

Benefits	User friendly, whole building simulations feasible, comparisons among simulation results and monitoring data possible	Due to its modular approach, TRNSYS is extremely flexible for modelling a variety of energy systems in differing levels of complexity	Powerful performance monitoring platform (building and HVAC) on a very detailed level (component level). Simaxx also monitors comfort conditions, data quality from GBS and working of HVAC systems	import and export using IFC
Relevancy to INSITER	Analysis impact of deviations (only those which correspond to the LOD of the input)	Analysis impact of deviations (only those which correspond to the LOD of the input); Development of submodels for testing HVAC-components	Performance monitoring for existing buildings (use phase); simulation data can be imported for easy comparison with monitoring data	For simulating error's impact on building performance.
Challenges	Limited modelling of concepts with natural ventilation, energy plants. Closed commercial package, physical models are not public available	The interface is not user-friendly	Not suitable for inspection of new building during the realization phase as Simaxx is not installed yet	Compared to other software, RIUSKA has more functional advantages. However, the software doesn't belong to the best ones. The most critical issue is that RIUSKA is based on DOE2 which is not fully proven

Interoperability issues	In a BIM workflow, VABI Elements derives the building geometry and room names from the BIM model (Figure 5.1). Geometry adjustments in the BIM model can quickly be exported to VABI. In current practice (the main) part of the required input, such as the physical material properties, building function and HVAC installation, is specified manually. A few possibilities exist for bi-directional communication with the BIM-model.	In a BIM workflow, the building geometry is derived from a SketchUp model. Transfer to the TRNSYS-environment is not user-friendly. A lot of details of the HVAC-components are not well specified in formats like IFC and have to be added manually. TRNSYS has also interfaces with various other simulation packages such as COMIS, CONTAM, EES, Excel, FLUENT[7], GenOpt[8] and MATLAB	No connection to the BIM model	good connectivity to BIM
Reference	http://www.vabi.nl/Producten/Vabi-Elements/Vabi-Elements-2.0 http://www.board.com/en/capabilities-dashboarding	http://www.trnsys.com/ http://www.board.com/en/capabilities-dashboarding	http://www.simaxx.com/	http://www.granlund.fi/en/software/riuska/

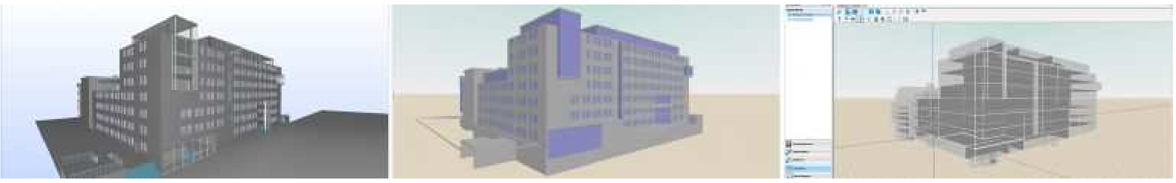


Figure 5.1 VABI building geometry derived from BIM model [source: vabi.nl]

TRNSYS

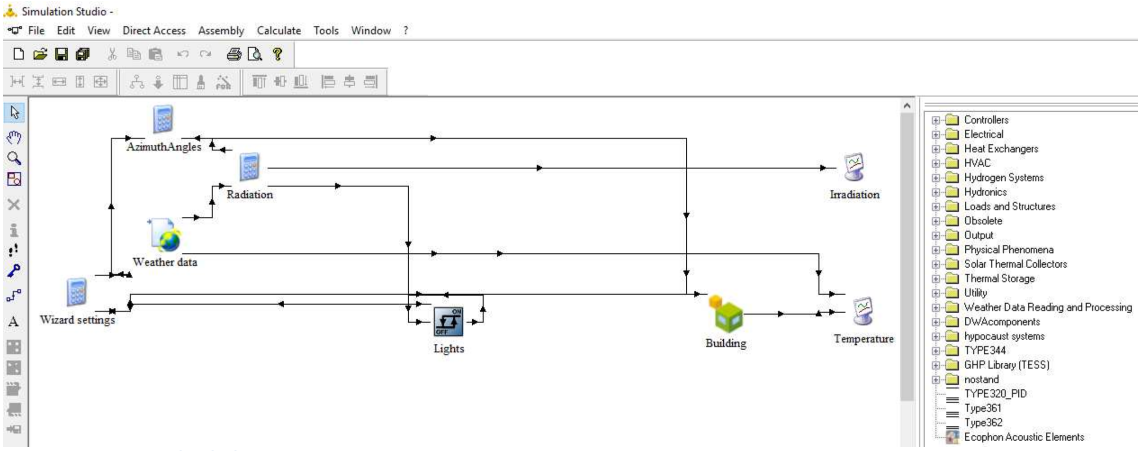
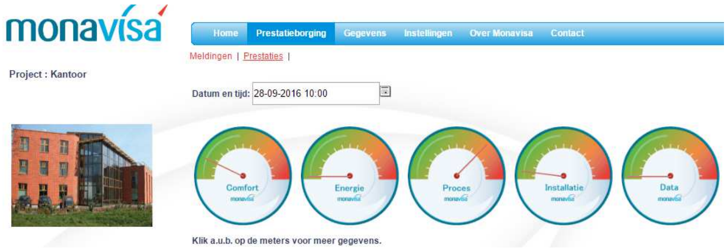


Figure 5.2 TRNSYS Simulation studio



Project : Kantoor

Datum en tijd: 28-09-2016 10:00

Klik a.u.b. op de meters voor meer gegevens.

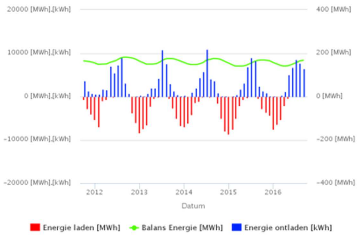


Figure 5.3 Simaxx performance dashboard

Figure 5.4 Simaxx monitoring visualization

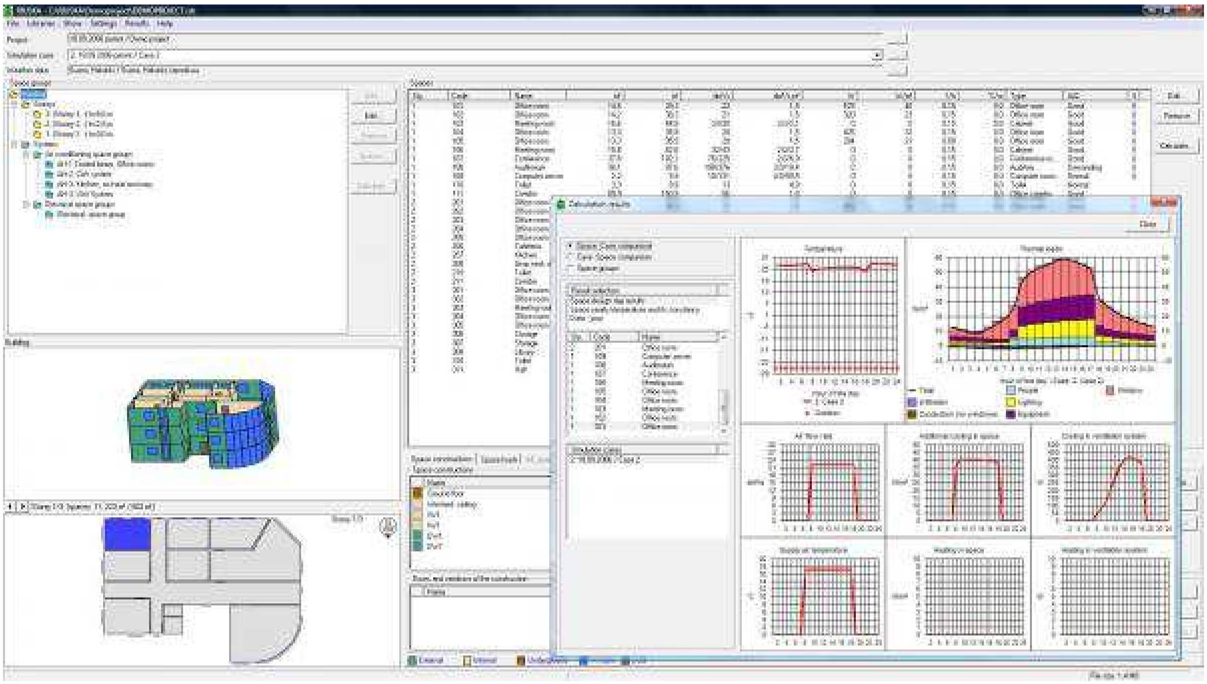


Figure 5.5 RIUSKA screenshot [source: http://riuska.nl]

<u>INSITER decision for software development:</u>

The analysis shows that until now, the best performing energy performance software has limited inoperability with BIM: import of the building geometry in an IFC format is possible in VABI and TRNSYS, but most building details, such as material properties and HVAC components, have to be specified manually.

For the time being, the chosen solution is:

- Vabi Elements for building performance calculation due to its user friendliness and BIM connectivity;
- TRNSYS for HVAC performance simulation due to its powerful simulation possibilities, and;
- Simaxx for building and HVAC performance monitoring due to available knowledge and experience.

The challenge for the INSITER software will be to export BIM models that can easily be imported to VABI or TRNSYS. In addition, the software should specify the building properties in a clear way to achieve a quick set-up of the energy performance simulation models. Unfortunately, the simulation models do not (yet) offer options for bi-directional communication with the BIM model, so that simulation results can be exported to the INSITER software. This option will be discussed with the simulation software developers.

6. Software architecture

The members of WP4 have elaborated the overall architecture including necessary databases and servers. The architecture design is shown in D3.1 in chapter 6 (adaptation layer BIM architecture).

6.1 BIM server

The picture below illustrates the relationship between the actors (with their device), the functional entity that provides INSITER services to these actors and a description of the interaction between the actors and the functional entities. The direction of the arrows can be read as follows:

- direction from functional entity to the actor indicates that the actor receives information from the functional entity,
- direction from actor to functional entity indicates that the actor provides information to the functional entity (but might receive information as well).

Note that besides the extended BIM server functionality there is also the Augmented Reality App running on the mobile device itself. The extended BIM server functionality refers only to the backend server part of the AR tool.

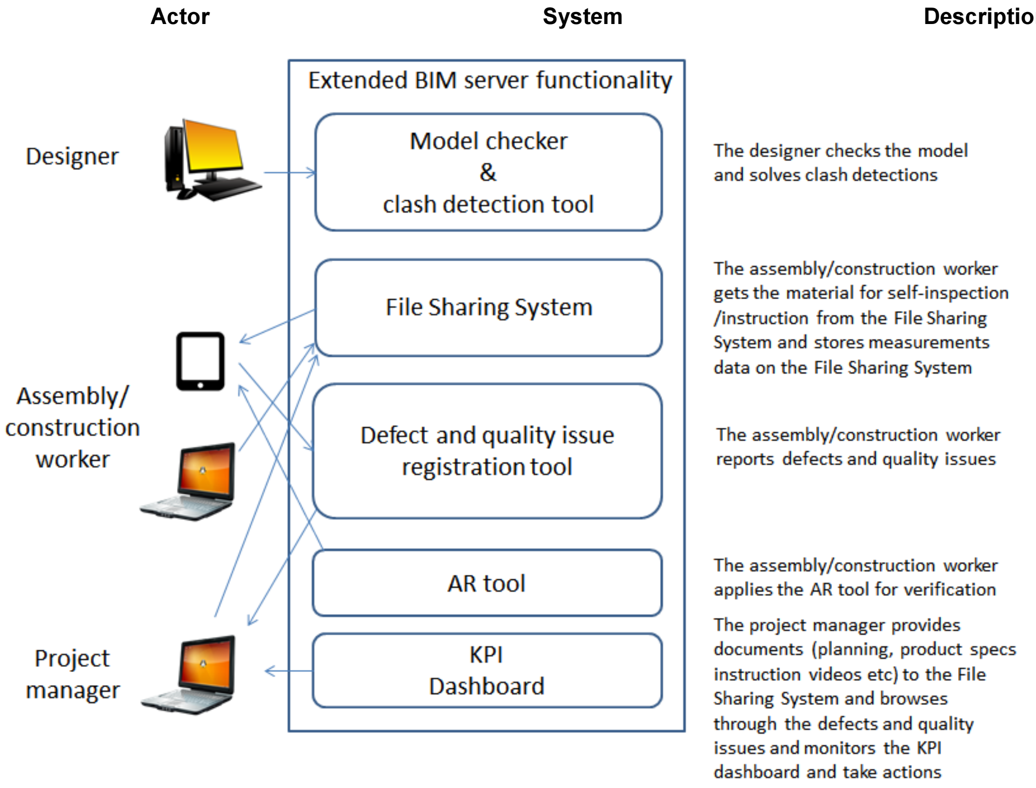

Actor
System
Description
Extended BIM server functionality
Designer
Model checker & clash detection tool
The designer checks the model and solves clash detections
File Sharing System
The assembly/construction worker gets the material for self-inspection /instruction from the File Sharing System and stores measurements data on the File Sharing System
Assembly/ construction worker
Defect and quality issue registration tool
The assembly/construction worker reports defects and quality issues
AR tool
The assembly/construction worker applies the AR tool for verification
Project manager
KPI Dashboard
The project manager provides documents (planning, product specs instruction videos etc) to the File Sharing System and browses through the defects and quality issues and monitors the KPI dashboard and take actions

Figure 6.1 Extended BIM server functionality

6.2 Databases

For the purpose of the INSITER toolset, three databases will be used:

- **Open Source BIM server**: is selected as alternatives, like BIM+, do not offer much more. Besides, consortium partners, CARTIF and RDF, have knowledge about this OS BIM server.
- **Mongo DB** (was PostGreSQL): this database is very suitable for dynamic data; a SQL based server is not the best solution.
- MS **SharePoint**: this solution will be used for all static files not being IFC.

Each database within the INSITER toolset solution is an existing mature product; this means a well-defined API is available for all three databases used. Both OS BIM server and Mongo DB are open source products and their API is open, clear and extendable. SharePoint is a closed source Microsoft product, however it supports CMIS, what is the open standard for interaction. The collective of all three API's will be defined as the INSITER API, the first applications will access the individual databases directly, in a later stage a common API will be added based on the experiences found during the first development phase.

Portable devices will have applications that access the databases directly, both RE Suite and IFC files will directly put content within the OS BIM server and different tools like Augmented Reality tools will be able to receive their content directly from the databases.

Figure 6.2 illustrates the three databases and related connections. However, databases will be target for D3.4. Final decision will be taken in the next period.

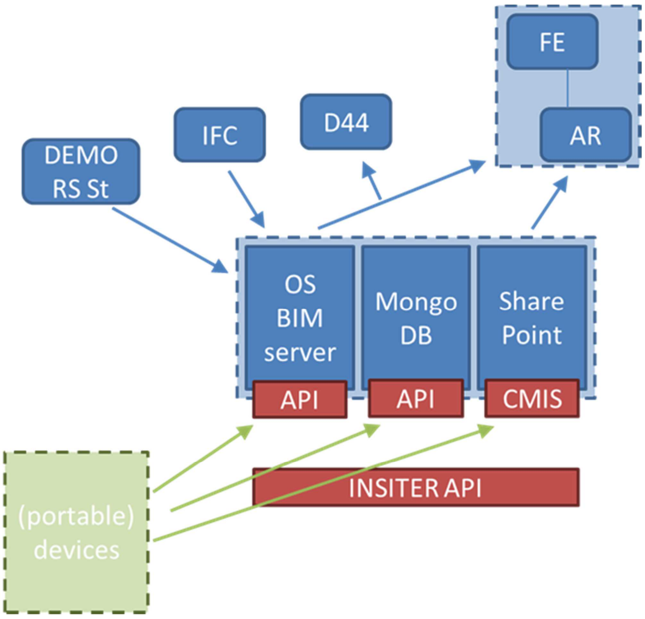


Figure 6.2: databases for INSITER toolset

6.3 Architecture of the INSITER toolset

Based on the discussion with WP1, WP2 and WP4 is the architecture of the whole INSITER toolset designed. This design will be the basis for further development (resulting in D3.4).

The INSITER toolset exists of an INSITER BIM platform containing 3 different databases as described above. Each database has its own API and all API's are combined in the INSITER API. Individual applications used and/or developed or extending within the project are directly or indirectly communicating with the INSITER BIM platform. In above picture we find the landscape of all different relevant applications, the colours on the connectors to the INSITER BIM platform show the databases affected by or used by the individual solutions. The landscape is divided in applications used on-site or off-site and in what layer the solution is defined, the layers are described in the BIM chapter and a more detailed description can be found in D4.3.

Figure 6.3 illustrates the whole architecture design.

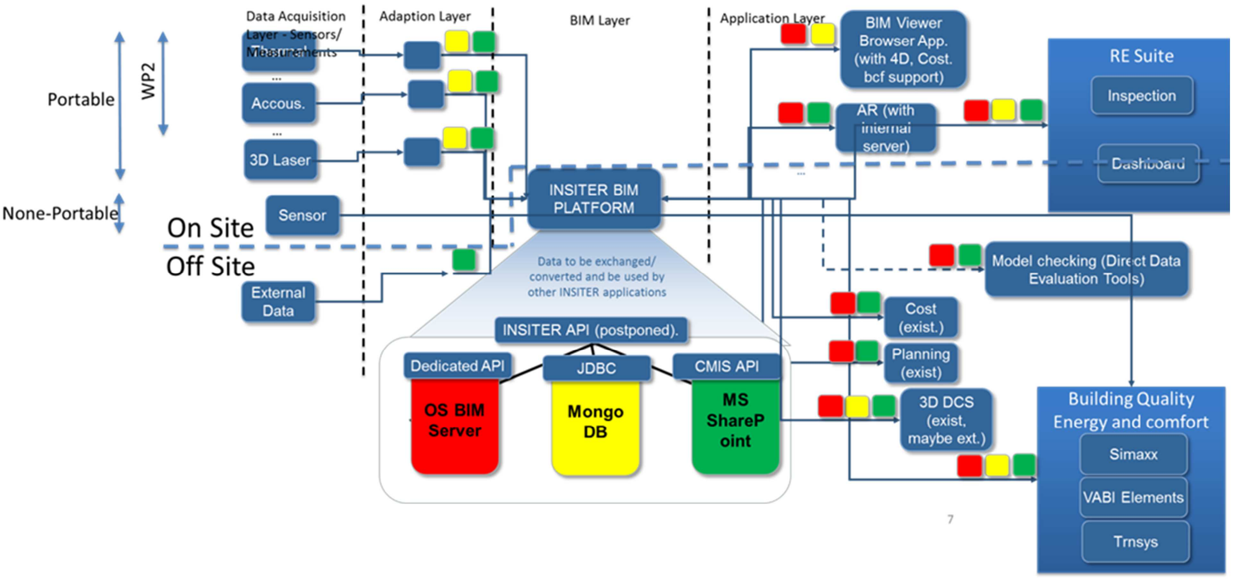


Figure 6.3: architecture of the whole INSITER toolset

7. Software proof-of-concept

The INSITER toolset will not be developed from scratch. Elements will be (re)used from the existing software packages as listed below:

- DEMO RE Suite software (RE Dashboard, RE Maintenance)
- Desite MD for model checking and viewing
- FHG Instant3Dhub platform for data visualization as well as the VisionLIB technology for computer vision based tracking as basis for AR software development
- Microsoft SharePoint (document management)
- Simaxx (performance and energy monitoring platform for building and HVAC systems)
- Navisworks + Plug-In for clash detection
- Open source BIM server
- Several BIM models e.g. a hospital in Cologne
- TRNSYS (energy performance simulation)
- Vabi-elements (energy performance simulation)

The enhancements required to make the INSITER toolset work are listed in the next section

7.1 Software to be developed

Inspection tool (App) for quality and defect registration:

- Navigation possibilities for selecting a BIM part via a BIM viewer
- Presenting requirements to the worker
- Registration non compliances (deviations) that are encountered during inspections
- Presenting additional information related to self-instruction and self-inspection to the worker

The App communicates with the BIM server

Augmented reality prototype application and its functionalities:

- web-based visualization client that support computer vision based tracking
- an interface in order to apply the INSITER BIM model as input for AR functionality
- click functionality on (parts of) the design model on the mobile device to derive additional information
- adaption of tracking technology to dedicated INSITER use case

Beside the 3D BIM models, additional content is planned be visualized in the AR demonstrator (see page 29)Moreover, the AR software will enable the access of other external planning documents (e.g. PDF) on file share systems (e.g. SharePoint) linked within the BIM model (part/object) as references.

The augmented reality application will be realized within a dedicated demonstrator use case environment with sufficient geometrical structures in the area to be tracked. The demonstrator setup will take place within a dedicated building environment or construction site, such as the hospital building in Cologne, Germany where the BIM model is available and access to the real building is possible.

Condition assessment app functionalities:

- an interface in order to apply the INSITER BIM model as input and storage for condition assessment (e.g. construction list)

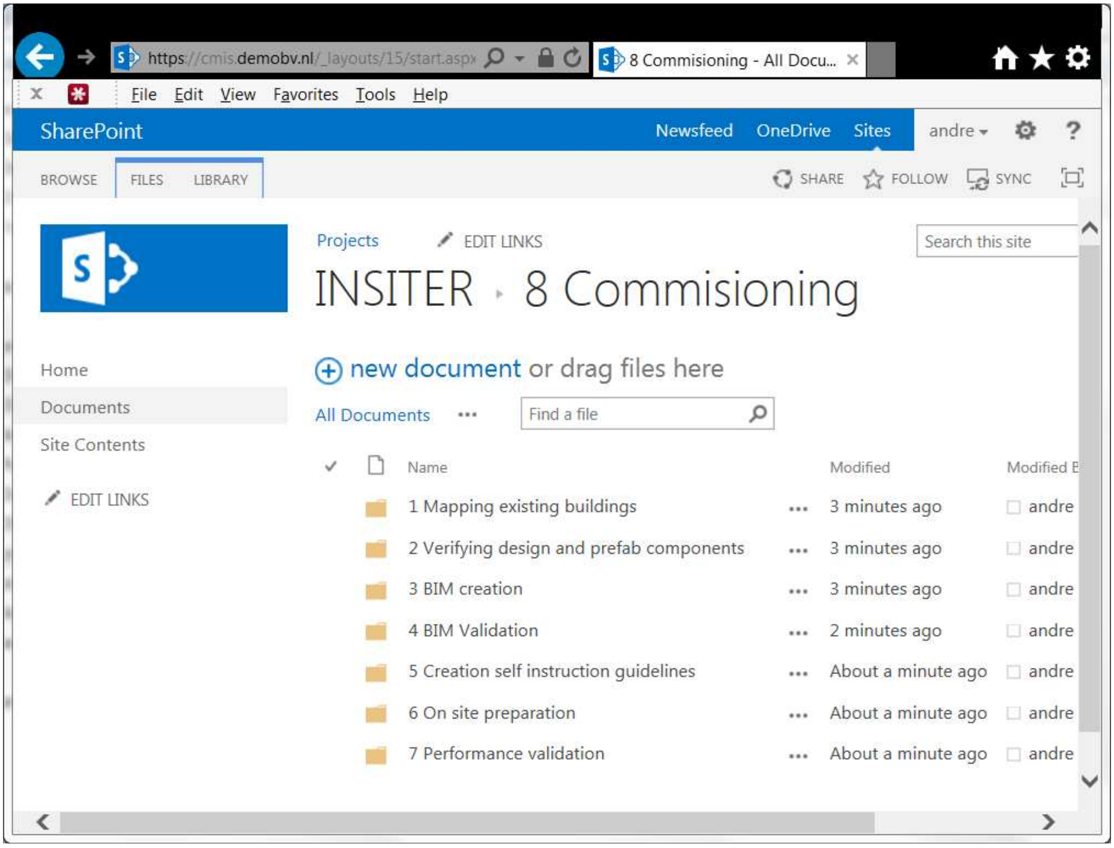
Energy assessment app functionalities:

- an interface in order to apply the INSITER BIM model as input and storage for energy assessment (e.g. construction list)

BIM platform functionalities (as relevant for D3.3):

- Central storage of the design model
- Central storage of the related process, product and project files and measurements data

The software that is going to be developed is not software that has to be written from scratch on. The required functionality is developed on top of existing software that is already operational and running. The consortium partners have foreseen that this existing software (proven technology) has a sound basis to build the required functionality. The scheme of software architecture (BIM) has pointed out what the structure and interrelationship of the software components are. Figure 7.1 shows the proof of concept that will serve as the basis for the final software. The model and checking and clash detection tools are covered in other INSITER parts and therefore not illustrated in this deliverables.



INSITER › 8 Commisioning

Name	Modified	Modified B
1 Mapping existing buildings	3 minutes ago	andre
2 Verifying design and prefab components	3 minutes ago	andre
3 BIM creation	3 minutes ago	andre
4 BIM Validation	2 minutes ago	andre
5 Creation self instruction guidelines	About a minute ago	andre
6 On site preparation	About a minute ago	andre
7 Performance validation	About a minute ago	andre

Figure 7.1: File sharing system based on open standard CMIS

This SharePoint server with CMIS open standard interface is the backbone of the information and data exchange. It is a very powerful and flexible toolset for this purpose.

The defect and quality issue registration tools (apps and applications) are based on the RE Suite and Simaxx. Some examples for proof of concept are shown in Figure 7.2.

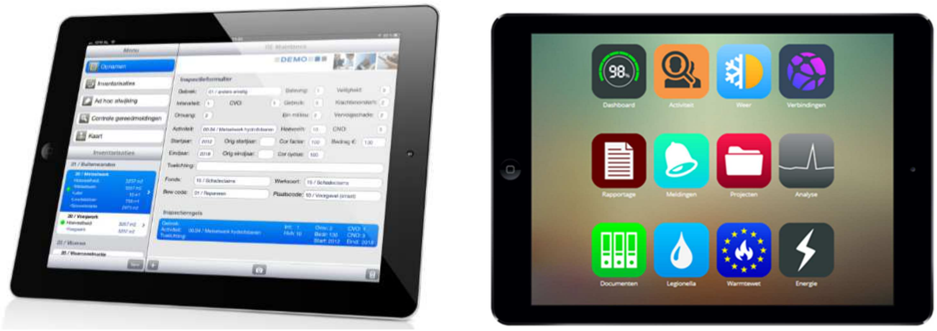
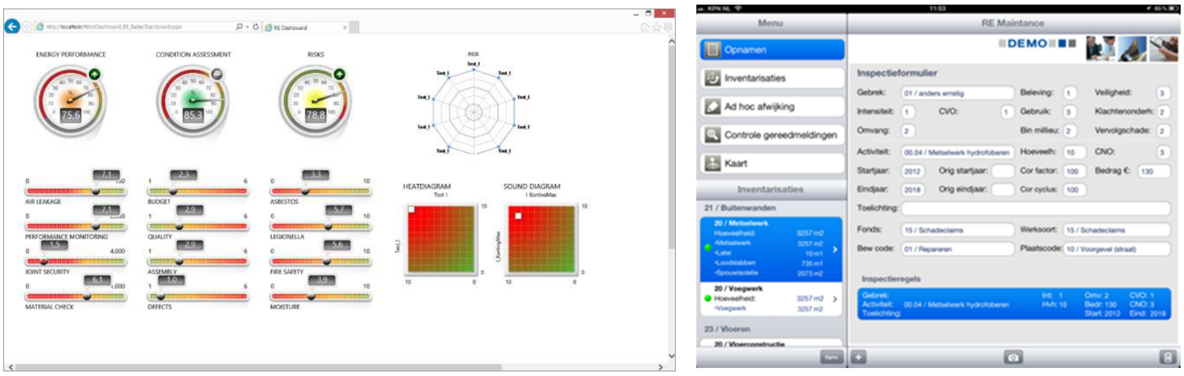


Figure 7.2: RE Suite and Simaxx tools

The registration of defects and quality issues is handled by the software application and shown to the several workers in so called KPI dashboards.



Inspectieformulier

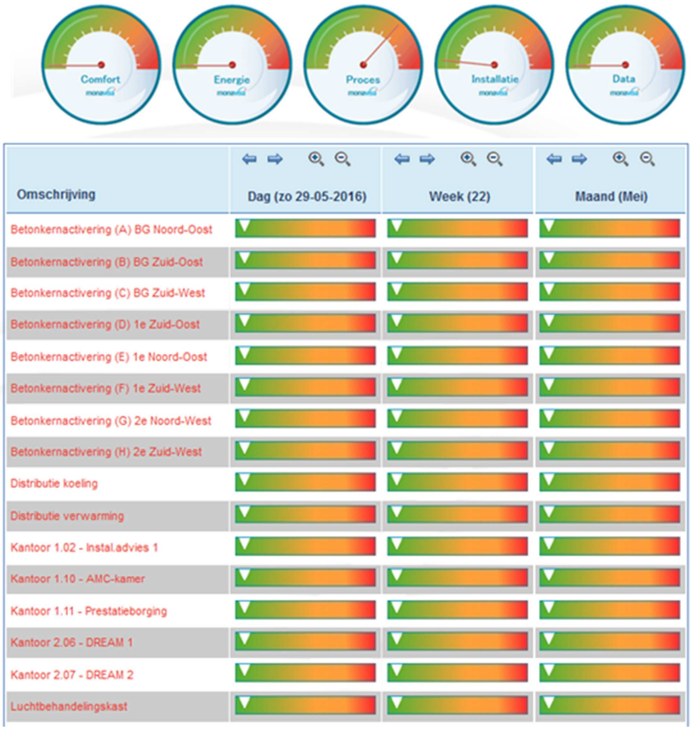
Gebrek: 01 / anders ernstig
Beleving: 1
Veiligheid: 3
Intensiteit: 1
CVO: 1
Gebruik: 3
Klachtenonderh: 2
Omvang: 2
Bin milieu: 2
Vervolgschade: 2
Activiteit: 00.04 / Metselwerk hydrofoberen
Hoeveelh: 10
CNO: 3
Startjaar: 2012
Orig startjaar:
Cor factor: 100
Bedrag €: 130
Eindjaar: 2018
Orig eindjaar:
Cor cyclus: 100
Toelichting:
Fonds: 15 / Schadeclaims
Werksoort: 15 / Schadeclaims
Bew code: 01 / Repareren
Plaatscode: 10 / Voorgevel (straat)

Inspectieregels

Gebrek: Int: 1 Omv: 2 CVO: 1
Activiteit: 00.04 / Metselwerk hydrofoberen Hvh: 10 Bedr: 130 CNO: 3
Toelichting: Start: 2012 Eind: 2018

Figure 7.3: KPI dashboard

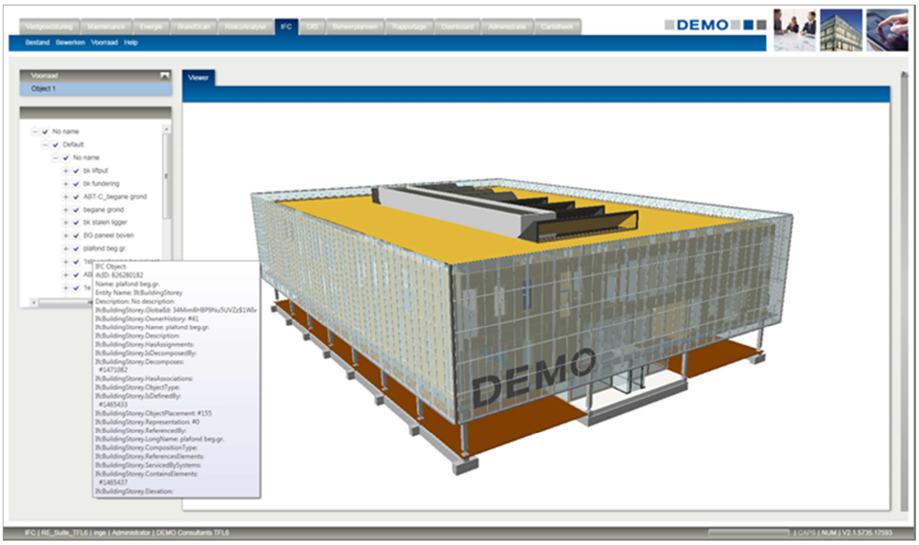
For energy performance monitoring there will be developed special dashboards based on Simaxx.



Omschrijving	Dag (zo 29-05-2016)	Week (22)	Maand (Mei)
Betonkernactivering (A) BG Noord-Oost	▼	▼	▼
Betonkernactivering (B) BG Zuid-Oost	▼	▼	▼
Betonkernactivering (C) BG Zuid-West	▼	▼	▼
Betonkernactivering (D) 1e Zuid-Oost	▼	▼	▼
Betonkernactivering (E) 1e Noord-Oost	▼	▼	▼
Betonkernactivering (F) 1e Zuid-West	▼	▼	▼
Betonkernactivering (G) 2e Noord-West	▼	▼	▼
Betonkernactivering (H) 2e Zuid-West	▼	▼	▼
Distributie koeling	▼	▼	▼
Distributie verwarming	▼	▼	▼
Kantoor 1.02 - Instal.advies 1	▼	▼	▼
Kantoor 1.10 - AMC-kamer	▼	▼	▼
Kantoor 1.11 - Prestatieborging	▼	▼	▼
Kantoor 2.06 - DREAM 1	▼	▼	▼
Kantoor 2.07 - DREAM 2	▼	▼	▼
Luchtbehandelingskast	▼	▼	▼

Figure 7.4: Simaxx dashboard

The backbone of the software's data exchange is the open BIM standard IFC. The several software components will use these data files, or federation of data files through the CMIS server. The needed functionalities therefor will be leaning on the communication through IFC and comprise visualization, calculation, analysis and modifying. An example of this software and of how this will look like is shown below.



IFC Object
AiID: 826280182
Name: plafond beg.gr.
Entity Name: IfcBuildingStorey
Description: No description
IfcBuildingStorey.GlobalId: 34MimBHBP9Nu5UVZz$1Wlv
IfcBuildingStorey.OwnerHistory: #41
IfcBuildingStorey.Name: plafond beg.gr.
IfcBuildingStorey.Description:
IfcBuildingStorey.HasAssignments:
IfcBuildingStorey.IsDecomposedBy:
IfcBuildingStorey.Decomposes:
#1471082
IfcBuildingStorey.HasAssociations:
IfcBuildingStorey.ObjectType:
IfcBuildingStorey.IsDefinedBy:
#1465433
IfcBuildingStorey.ObjectPlacement: #155
IfcBuildingStorey.Representation: #0
IfcBuildingStorey.ReferencedBy:
IfcBuildingStorey.LongName: plafond beg.gr.
IfcBuildingStorey.CompositionType:
IfcBuildingStorey.ReferencesElements:
IfcBuildingStorey.ServicedBySystems:
IfcBuildingStorey.ContainsElements:
#1465437
IfcBuildingStorey.Elevation:

Figure 7.5: visualization of BIM

7.2 Energy Performance influence after found deviations during inspection

Energy Performance is a typical area where BIM can play an important role. As BIM not only contains the pure geometry of objects, but also their interrelations and properties. A BIM model contains therefore not only the geometry of a wall, a floor, a roof and even a space, but also the knowledge it is a certain object like wall, is surrounding the space and has certain values for energy performance like for example the energy coefficient.

There are many reasons why there is only limited use of such real BIM information in energy performance calculations, most importantly:

- BIM models are not created or not exported in an open format, either because the simply knowledge how to do this is missing or because the BIM model is so detailed and contains so much of the knowledge put in by the architect that he fears of having the design in hands of competitors.
- Another reason for not distributing the BIM model if available is the responsibility of issue in case the model is incorrect. As the information is so detailed also the room for errors and mistakes is increased, especially concerning typical energy discipline specific knowledge like energy coefficients.
- Complex local regulations are also an important reason for missing support for BIM in Energy calculation applications. The algorithms often have very specific requirements of information not or only implicitly available within BIM. For example a wall with a thickness of 30 cm existing of 4 different layers is very common, the area between a room and its neighbouring rooms is strongly depending on what layer can be included in the room space and what layer should be really assigned to the wall, in many cases the IFC model has different measures for space boundaries because the wrong layers are allocated to the wall.
- Together with the previous reason the complexity of an implementation is also a reason for not having BIM in the energy applications.

As component provider of BIM enabling libraries RDF has cooperated in implementing BIM support in a limited set of applications for energy calculations, i.e. the early adopters. One of these applications is VABI Elements, also used by partner DWA within the project. The BIM / IFC support in this application is advanced and mature what makes it possible to use it for INSITER.

Energy calculation based on BIM models in the open format IFC is new and only a limited set of applications can be used to make this possible. This also requires a strict set of modelling guidelines improving the results and minimizing the effort to complete the information by hand in the energy calculation software. In the follow-up deliverable D3.4, this step will be elaborated.

In order to avoid confusion or misinterpretation, it should be clarified that in INSITER, the energy calculation based on BIM will ***not*** be used to calculate the ***expected*** energy use of the building, but to ***identify and quantify the impact of flaws or defects*** found during inspection which negatively affect the ***overall energy performance***.

The use of a BIM model in INSITER and the capabilities of BIM enabled energy calculation software, i.e. the early adopters, allows in theory valuable feedback during the inspection process about severity of consequences of found deviations from the design.

The following steps are required:

- A BIM model according to modelling guidelines for the energy calculation
- The inspection
- Store the inspection deviations within the inspection tool as described above
- Adjust the BIM model slightly to actualize the deviations found
- Run the energy calculation application and compare the overall results with the model without the deviations

Applying this new process is not as easy as it sounds. There are two major complexities within the process:

- How to actualize the BIM model based on the identified deviations during inspection
- Minimize the hand input during a run of the energy calculation application.

How to actualize the BIM model based on the identified deviations during inspection

During the inspection process the deviations from the developed and available BIM model are described. In many cases these deviations are clear and result in clear actions; however in some cases fixing the deviations can be expensive, dangerous or complicated due to other reasons. In the cases where fixing the deviations is complicated this solution will give an indication of the expected impact of the deviation on the energy performance of the overall building.

This means the noted deviation needs to be embedded in the BIM file. Technically it means the BIM model needs to represent the problem. Technically this will result in the request to duplicate to show alternatives in separated objects where one of them has difference in behaviour compared to the design. In here we need to take into consideration the limitations of the energy calculation application. Still the solution is expected to be generic set of operations on top of the existing IFC / BIM file.

Minimize the hand input during a run of the energy calculation application

Although using BIM has major benefits even in the mature and advanced solution of VABI a lot of extra hand-input by specialists is needed to complete the missing information that either was missing in the BIM model or was too complicated to derive from the BIM model. In the standard process this is no problem as the alternative is to put all data in by hand; using BIM for automatically derived information gives already an enormous reduction of work.

Within the INSITER solution however we will run the same process several times; therefore adding (often the same) information after loading the BIM model is time consuming, error prone and costly. This is therefore one of the 2 main issue that needs to solved within implementing this process.

8. Conclusions and follow-up software development plan

This chapter describes the conclusions of this deliverable and the consequences for the follow-up work in this Task 3.2 and D3.4.

8.1 Feasibility analysis

8.1.1 Feasibility analysis Self-inspection related requirements

The self-inspection related requirements will be target for the next deliverable. It involves enhancement of the following software:

- AR software with the functionalities as described under section 4.3
- Model checking software to operate with BIM data of the Open BIM Server
- Clash detection software to operate with BIM data of the Open BIM Server
- Import software to operate with BIM data of the Open BIM Server

These requirements are fully implementable.

8.1.2 Feasibility analysis Self-instruction related requirements

The self-instruction related requirements will be target for the next delivery. It involves configuration of the next software:

- SharePoint with the functionalities as described under section 4.1

These requirements are fully implementable.

8.1.3 Feasibility analysis Process related requirements

The process related requirements will be target for the next delivery. It involves enhancement of the next software:

- SharePoint with the functionalities as described under section 4.2
- RE Dashboard software to operate with BIM data of the Open BIM Server as described under section 4.2

These requirements are fully implementable.

8.2 Scope

This section describes the scope for the software development in the next phase.

Important here is to develop software that is able to demonstrate the INSITER concept hereby carried out within the limited project time and budget. The scope is defined as follows:

1. At least three BIM models (one for new buildings, one for refurbishment and one for maintenance) coming from WP4.1 will be supported. For AR only 1 BIM model will be applied;
2. At least three document types for self-inspection (product specs, planning, and checklist) will be supported;
3. At least two document types for self-instruction (instruction document, instruction movie) will be supported;
4. At least three measurement devices (e.g. thermal/imaging, humidity, and acoustics/vibration) will be applied;
5. Energy performance measurements simulated using tools that are on the market like Vabi Elements, TRNSYS, and Simaxx;
6. The set of KPIs as defined by WP1 will be supported.

8.3 Input to be received from other work packages

In this section is described what input is expected from the other work packages in order to build a demonstration case within the next phase of the INSITER project.

Input expected from WP1:

During the building quality and energy performance assessments a checklist needs to be provided to the person responsible for performing these tasks. The checklist will be digitalized so that it can be applied in a registration toolset on-site. Checklists (template) and documents (types) are required for:

1. Inspection of incoming goods on-site (reception of materials)
   e.g. for completeness check (against bill of material and delivery note in case of partial deliveries) and check if the product is damaged/proper size or not. Besides the checklist, the required documents are for example: BIM, delivery note, product specs to be defined by WP1.
2. Inspection of process related activities on-site (activity inspection)
   e.g. a checklist per activity type and definition of documents that is relevant per activity.
3. Execution of planned tests on-site
   e.g. a checklist/test list and definition of documents that is relevant per type of test.

In addition WP1 should deliver the input for developing electronic forms for the registration of defects and quality issues.

Input expected from WP2:

During self-inspections thermal/imaging, acoustic/vibration and 3D laser scanning measurements may be applied. In order to collect, convert and import the measurement data into the BIM server it is important to receive from WP2 the actual file formats and middleware solutions to enhance the files with 3D geometry info. With this information import tools can be developed for the BIM server.

AR software with the following functionality:

- Retrieving BIM model
- Overlay the reality as seen through the camera with the BIM model
- Click functionality to click on parts of the BIM model and access the linked information registered for that part

Input expected from WP4:

BIM model and a Model Checking and Clash Detection methodology to apply on design models stored on the BIM server.

8.4 Decisions and conclusions

Concluded is that within the scope of INSITER, the project will make as much as possible advantage of the already existing software brought in by the consortium partners and enhance it in order to fulfil the needs. This will not only prevent additional costs but it will also be easier to make the required enhancements to the software. It applies to the FHG Instant3DHUB and computer vision based tracking solutions for AR development, RE Suite dashboard and condition assessment software from DMO and Navisworks plug-in software from HVC. For document management related software Microsoft SharePoint is chosen because it is already in use by several partners, knowledge is available and it is perfectly suitable to handle the document related requirements.

8.5 Next steps in Task T3.2 / Deliverable D3.4

This functional program of requirements results in a set of software applications that can be used directly and a set of software applications that still need some development.

The ‘Defect and quality issue registration tool’ is one of the tools that still needs developments, however when available it will be beneficial for both building quality and energy assessment.

The fact that state-of-the-art energy calculation applications are adapting BIM and INSITER has a central BIM model available enables us to create decision support (concerning energy assessment) during or after the inspection process. Although the solution may seem logical there are 2 major issues to be tackled in the next period, i.e. converting the found issues into a correct computer understandable representation within the BIM model and the minimization of user-input during the energy calculations on the adjusted models.

9. References

[Microsoft - SharePoint]
Content management, document management, workflow management and web content management software
https://products.office.com/en-GB/sharepoint?legRedir=true&CorrelationId=461ba077-b342-4c98-8c6d-e2f995c6a145

[Fraunhofer - Instant3Dhub platform]
Visual computing platform as a comprehensive solution for interactive data visualization
https://www.igd.fraunhofer.de/en/Institut/Abteilungen/VCST/Projekte/Instant3DHub

[DEMO Consultants BV - RE Suite]
Software platform for Real Estate Information Management
http://www.demobv.nl/software_uk

[Ceapoint - Desite MD]
Visualizes combined BIM content from several sources
http://www.ceapoint.com/index.php?option=com_content&view=article&id=61&Itemid=55

[Autodesk - Navisworks]
Software package for holistically review of integrated models and data
http://www.autodesk.com/products/navisworks/overview

[Open - Source BIM server]
Storage for BIM related data accordance with the open standard IFC (IFC2x3 TC1 and IFC4 ADD1)
http://bimserver.org

[Vabi - Vabi Elements]
Building performance software platform
http://www.vabi.nl/Producten/Vabi-Elements/Vabi-Elements-2.0

[TRNSYS - TRNSYS]
Behaviour simulation of transient systems
http://www.trnsys.com/

[Simaxx - Monavisa]
Energy consumption monitoring software for buildings
http://www.simaxx.com/

[Grunland - RIUSKA]
Software for energy calculation and façade optimization
www.granlund.fi/en/software/riuska/

References to deliverables from other work packages in INSITER project:

- Di Giulio R. and WP1 Team (2015),"Best practises and existing shortcomings", deliverable report, INSITER D1.1
- Bonsma P. and WP3 Team (2016),"Functional program of requirements for planning and cost", deliverable report, INSITER D3.1
- Braun JD. and WP4 Team (2016)," BIM (aggregate models and whole models) of new and existing buildings including the construction sites", deliverable report, INSITER D4.1
- Bonsma P. and WP4 Team (2016),"Framework for BIM, hardware and software data interoperability", deliverable report, INSITER D4.3