COGITO

CONSTRUCTION PHASE
DIGITAL TWIN MODEL

cogito-project.eu

D7.8 – Construction Digital Twin 3D Visualisation Module v2

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958310

D7.8 – Construction Digital Twin 3D Visualisation Module v2

Dissemination Level:	Public
Deliverable Type:	Demonstrator
Lead Partner:	Hypertech
Contributing Partners:	UCL, UPM, DTU, UEDIN, NT
Due date:	31.10.2022
Actual submission date:	27.10.2022

Authors

Name	Beneficiary	Email
Vassiliadis, Michalis	Hypertech	m.vassiliadis@hypertech.gr
Tsalis, Christos	Hypertech	c.tsalis@hypertech.gr
Papafragkakis, Apostolos	Hypertech	a.papafragkakis@hypertech.gr
Giannakis, Giorgos	Hypertech	g.giannakis@hypertech.gr
Kakarantzas, Pavlos	Hypertech	p.kakarantzas@hypertech.gr
Papapolyzos, Ntinos	Hypertech	n.papapolyzos@hypertech.gr
Fotopoulou, Anastasia	Hypertech	a.fotopoulou@hypertech.gr
Charbi, Theofania	Hypertech	t.charbi@hypertech.gr
Lilis, Georgios	UCL	g.lilis@ucl.ac.uk
Katsigarakis, Kyriakos	UCL	k.katsigarakis@ucl.ac.uk

Reviewers

Name	Beneficiary	Email
Tsakiris, Thanos	CERTH	atsakir@iti.gr
Meusburger, Elias Bruno	RSRG	elias.meusburger@rsrg.com

Version History

Version	Editors	Date	Comment
0.1	Hypertech	15.09.2022	Table of Contents
0.2	Hypertech	21.10.2022	Draft version ready for internal review
0.3	CERTH, RSRG	26.10.2022	First draft internal review
0.4	Hypertech	27.10.2022	Review comments addressed
0.5	Hypertech	27.10.2022	Final version
1.0	Hypertech	27.10.2022	Submission to the EC portal

Disclaimer

Executive Summary

Construction sites constitute a very dynamic, highly evolving environment; one of the primary goals of the COGITO ecosystem is to allow all involved stakeholders to remain up-to-date – even while off-site, in terms of work progress, construction quality and potential safety issues. On that account, a Construction Digital Twin 3D Visualisation module is implemented, hereafter termed Digital Command Centre or DCC; the purpose of this document is to report on this work, performed as part of tasks T7.3 and mainly T7.4.

DCC is a 3D visualisation module which makes use of the data stored in the COGITO Digital Twin Platform to create an intuitive graphical representation of a construction site Digital Twin. It is based on the popular 3D game engine Unity and utilises other work done in parallel within T7.3 and T7.4, namely the development of a 3D BIM COGITO Unity library and an IFCtoMesh generator service. Due to its web-based nature, DCC allows the end-users to gain a remote - yet detailed - view of the construction site, regardless of device or operating system. More particularly, it allows a detailed view of the as-planned 4D data, i.e., the geometrical representation filtered by the work execution time schedule, as well as the as-built data, i.e., the geometry of what has actually been constructed/implemented in a particular time frame. Making use of other COGITO components, it allows the user to enable overlays reporting on the accomplished work progress, potential health and safety issues or hazards, the quality of the work performed and location/IoT data. It should be noted that there is no business logic involved in the execution of the application; it merely handles the rendering/visualisation of the various components/entities, i.e., new data is not generated or transformed throughout DCC. The latter is addressed by the DTP and its inter-connected tools (i.e., the WODM, QC, H&S and IoT modules).

In this report, the module's architecture is illustrated, including details on the technology stack/implementation tools used. Considering that DCC is a GUI-based visualisation tool, a thorough, step-by-step walkthrough of the user interactions is presented; for each interaction, a relevant application snapshot is provided. To ensure that both the stakeholder and the functional/non-functional requirements as defined in deliverables D2.1, D2.4 and D2.5 are addressed, a corresponding section is added; any assumptions and restrictions are also highlighted. This document builds on top of deliverable D7.7 to report on the efforts carried out since the former's release; emphasis is placed on the DCC's newly added features and functionality by means of actual walkthrough examples. As mentioned in the conclusions, this second iteration of the DCC application shall meet all the functional and non-functional requirements as set out in the relevant deliverables. Nevertheless, DCC is expected to become fully interoperable with the COGITO platform during task T8.1, when all inter-related tools are also expected to have reached their final version. As such, minor changes, adaptations or rework might be necessary before the component's final release.

It should be stressed that in order to gain a better insight on the inner workings of this module, it is <u>strongly</u> recommended that this deliverable is evaluated in tandem with deliverable D7.6 Data Transformation for 3D BIM Rendering v2.

Table of contents

List of Figures

List of Tables

List of Acronyms

Term	Description
API	Application Programming Interface
BS	Business Scenario
COGITO	COnstruction phase diGItal Twin mOdel
DoA	Description of Action
DCC	Digital Command Centre
DT	Digital Twin
DTP	Digital Twin Platform
GQC	Geometric Quality Control
GUI	Graphical User Interface
H&S	Health & Safety
HSE	Health, Safety & Environment
HSEM	Health, Safety & Environment Manager
IoT	Internet of Things
PM	Project Manager
QC	Quality Control
QM	Quality Manager
SM	Site Manager
UC	Use Case
VQC	Visual Quality Control
WODM	Work Order Definition and Management

1 Introduction

1.1 Scope and Objectives of the Deliverable

The tools developed throughout the COGITO ecosystem aim to provide continuous monitoring of construction projects and their involved processes; the modelling and presentation of such a dynamic environment is accomplished by means of a digital twin which constantly evolves to reflect the state of the physical assets, entities and processes. This enables the construction stakeholders to stay up-to-date regarding the state of the construction site in terms of work progress, quality and safety. To this end, a Construction Digital Twin 3D Visualisation module is implemented, henceforth termed Digital Command Centre or DCC; the purpose of this document is to report on this work, performed as part of tasks T7.3 and T7.4.

After a systematic investigation of the end-user requirements as documented in deliverable D2.1, it was decided that the DCC application should provide a visualisation of the DT model, available off-site (remotely). This shall enable the involved stakeholders to gain access on all data - directly or indirectly affecting the state of the construction site, such as the work progress, potential quality control issues and health/safety concerns. It should be noted that DCC's objective is solely to visualise the DT data; other tools developed within COGITO are responsible for importing/exporting/modifying the data. DCC renders the as-planned data (i.e., the 3D BIM model and schedule data) allowing to overlay the available, related as-built data and other real-time data/annotations (i.e., IoT data and annotations generated by the QC, H&S and Workflow tools). All data are retrieved after communication with the DTP; no business logic (e.g., data transformation/inference) is involved in the context of DCC. Ultimately, DCC's goal is to provide the stakeholders with an enhanced virtual-construction-site visit experience.

The development of DCC is fully in line with the requirements raised in the project's DoA and the end-user requirements - more particularly the related Business Scenarios BS-1 (Workflow), BS-2 (Quality) and BS-3 (Safety) and the related Use Case, UC-4.1. The main stakeholders involved (i.e., the module's end-users) shall be the Project Manager (PM), the Site Manager (SM), the HSE Manager (HSEM) and the Quality Manager (QM). The DCC's contribution regarding the three BS could be summarised as follows:

- BS-1: the stakeholders shall be able to retrieve up-to-date information regarding the construction site status and developments; there is virtually no need for (an otherwise time-consuming) document evaluation and/or communication with the responsible parties. Additionally, the remote access to real-time data offered by DCC can significantly improve decision making; data is available anywhere and at any time.
- BS-2: the DCC end-users are capable of performing a remote assessment on the quality of the completed work saving time and effort.
- BS-3: using DCC HSE issues and concerns can be identified and summarised, allowing the PM and HSEM to timely react/take mitigation measures.

Being an end-user module, DCC provides a GUI to interact with. Using the GUI, the user can browse through the various menus and options and select the ones applicable to the generation of the intended visualisation; the rendering itself is based on Unity, a 3D game engine, while the communication with the DTP to retrieve the data and perform the necessary format adaptation/transformation is accomplished using the IFCtoMesh Generator service and the COGITO Unity Library as described in deliverables D7.5 and D7.6.

To gain a better insight on the inner workings of this module, it is strongly suggested that this deliverable is evaluated in tandem with deliverable D7.6 Data Transformation for 3D BIM Rendering v2.

1.2 Relation to other Tasks and Deliverables

Table 1 depicts the relations of this document to other deliverables within the COGITO project; the latter should be considered along with this document for further understanding of its contents.

Table 1 - Relation to other COGITO project's deliverables

Del. Number	Deliverable Title	Relations and Contribution
D2.1	Stakeholder requirements for the COGITO system	Definition of the required interactions and functionalities to be offered by the COGITO 3D visualisation module, i.e., the DCC application, following an investigation and evaluation of the stakeholder (end-user) requirements and the target business scenarios.
D2.5	COGITO System Architecture v2	Second version of the COGITO architecture report including DCC's specifications and implementation details (hardware/software requirements, programming language(s), development status, functional/non-functional requirements, inputs/outputs, along with the format, method, endpoint and protocol for each data type and interface) as well as its relations with the other COGITO modules and tools.
D3.4	COGITO Data Model & Ontology Definition and Interoperability Design v3	Documentation of the second version of the ontologies used/developed throughout COGITO (also available online in the COGITO ontology portal).
D7.2	Digital Twin Platform Design & Interface Specification v2	Detailed report on the Digital Twin Platform's architecture, elaborating on its authentication (identity and access management), data ingestion (project and BIM management), data persistence (file storage, project database), data management and messaging layers; the above are involved in the COGITO Unity Library, which is in turn used by DCC in order to communicate with the DTP platform.
D7.6	Data Transformation for 3D BIM Rendering v2	Second version of the document describing the components involved in the Data Transformation for 3D BIM Rendering, including the components for retrieving, extracting, transforming and loading the BIM model in a 3D geometric representation suitable for integration with Unity applications, e.g., DCC.
D7.10	Digital Twin Platform v2	Detailed description of the second version of the DTP providing instructions on how data exchange is accomplished across the former and the different tools and applications.

1.3 Structure of the Deliverable

In Section 1 of the deliverable, an introductory presentation of the DCC module is provided, including an overview of its core functionality and objectives.

In Sections 2.2 to 2.6, the DCC architecture, its implementation, dependencies and features are elaborately described, including a note on the licensing scheme.

In Section 2.7 usage walkthrough examples are included, thereby presenting the functionality of the developed GUI.

In Section 2.8 to 2.10 aspects related to the requirements coverage, development and integration status as well as the assumptions and restrictions applicable to this version of the module are addressed.

Section 3 concludes this deliverable, providing a brief summary as well as remarks on the planned future work.

1.4 Updates to the first version of the Digital Twin 3D Visualisation Module

This document builds on top of deliverable D7.7 Construction Digital Twin 3D Visualisation module v1; since the former's release, the following updates shall be reported:

- Transition of the DCC's runtime target to WebGL to allow access to the application through any web browser independent of platform or device
- Design of a web front-end (GUI) responsible for serving the application
- Rework of the application's GUI (menus, filters, panels)
- Rendering optimisations (including visual effects addition/improvements)
- Continued the integration with the DTP in tandem with the developments in COGITO Unity Library
- Improved/reworked overlays features and functionality based on the progress of the related tools

2 Digital Command Centre Application

The Digital Command Centre or DCC is a web application that facilitates the visualisation of a construction project as modelled and stored within the COGITO's Digital Twin Platform (DTP). Utilising state-of-the-art 3D visualisation technologies, it allows the user to remotely monitor all parameters of a construction project, from the as-planned data (building geometry, schedule) to real-time data such as IoT sensor readings, quality control checks, work progress and health & safety issues. The application serves as a read-only client for the DTP platform, i.e., it solely consumes data from it. A snapshot of the DCC tool upon loading a construction project in a web browser is provided in Figure 1.

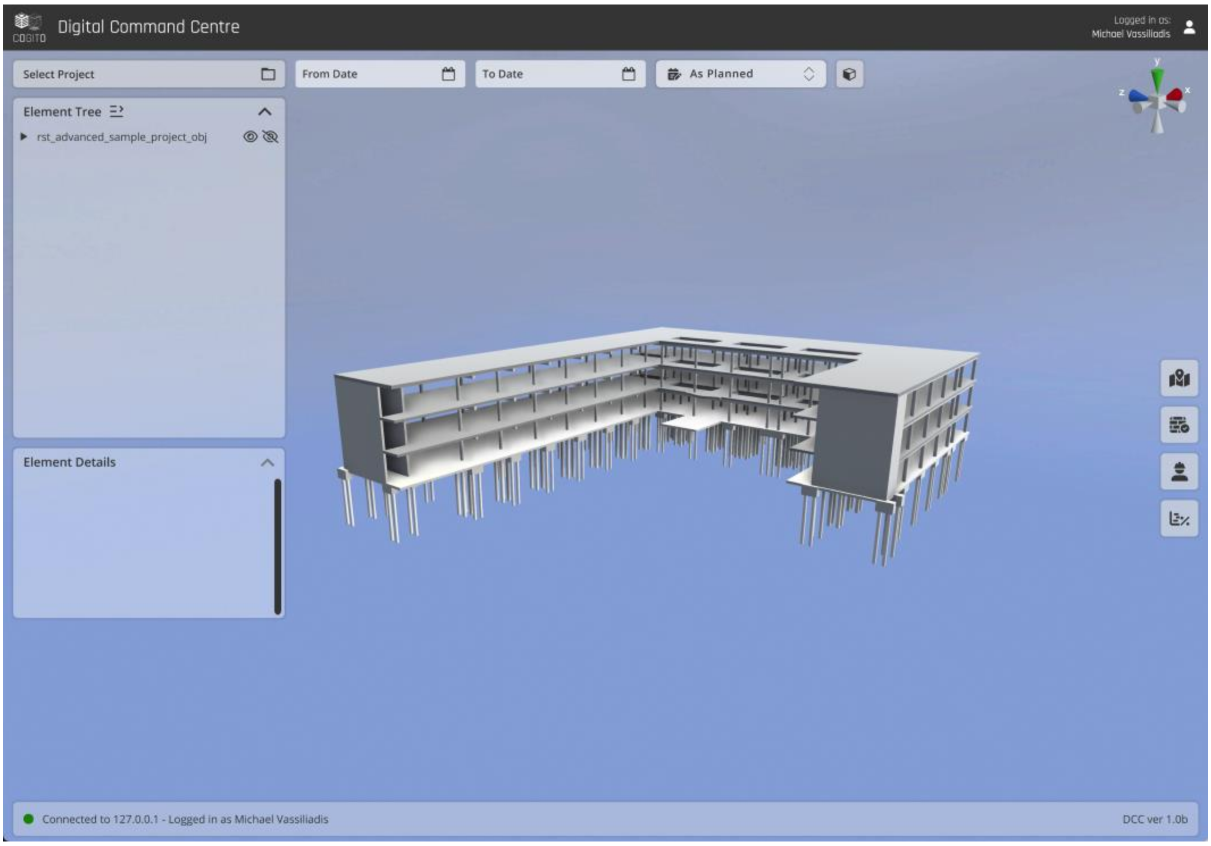


Figure 1 – Snapshot of the DCC upon loading a project in the browser

DCC is developed around the Unity framework and is responsible for:

a) providing a user-friendly, easy-to-use, web-based Graphical User Interface (GUI). The GUI shall enable:
  - user authentication
  - project selection
  - time-interval filtering (adjusting the data to be displayed)
  - the option to display (render) as-planned or as-built data
  - the display of related attributes and meta-data per-element
  - the display of the elements' hierarchy using a tree-view (also allowing BIM elements to be displayed or hidden)
  - the selection of overlays to be displayed (one at a time); the overlays correspond to the real-time data/annotations provided by the WODM, IoT, H&S and QC COGITO tools

b) generating the 3D scene and rendering the geometry corresponding to the (sub)set of the 4D BIM data (i.e., 3D geometry + schedule) selected through the GUI as well as to any enabled overlay. The user should be able to:
  - control the camera (rotate and pan the scene, zoom in/out, return to the original view)

- select a box-filter view
- select any element(s) (also reflected in the element-tree view and meta-data info)

A detailed walkthrough of the different functionalities offered in DCC is provided in Section 2.7.

2.1 Unity

The developed application is based on Unity, a 3D game engine platform. Unity enables the creation of complex 3D applications without requiring a dive into the low-level 3D rendering specifics; at the same time, it can guarantee optimal performance and portability of the application. In brief, Unity is a hybrid application development environment allowing both visual and programmatical construction of 3D scenes, elements and behaviours and is widely popular with regards to game development and architectural visualisation. A brief introduction to Unity as well as the rationale behind its selection as an implementation platform is provided in the following subsection.

Unity comprises of a development environment, namely the Unity Editor, as well as a scripting API. The Unity Editor allows the developers to create 3D objects, cameras, lighting, animations and place those inside scenes. One could think of a Unity scene as a level in a video game. Having created a scene with the required content, the API allows the developer to interact and modify the real-time behaviour of these elements, or even create new elements. This is done via Unity scripting.

2.1.1 Scripting

Even though Unity is written internally in native C/C++, its scripting engine uses C# or JavaScript. The developer has the option to choose between the two languages for every script created. The scripts allow the creation or modification of elements within a scene; as an example, a script can capture keyboard and mouse (or touchscreen) input and move the camera accordingly. The user interface fits in the same mode of operation. All elements within a scene are represented by the same data structure base, regardless them being game objects (e.g., an excavator), visual elements (i.e., the camera or a spotlight), or e.g., a UI button. Nevertheless, based on their type, these elements have different characteristics in terms of rendering by the engine. As an example, an excavator is a 3D object, whereas a UI button is a 2D object rendered in front of everything else. In the context of the API though, all those elements constitute GameObjects and expose the same basic features, such as the position in the 3D space, the size, rotation, material and others.

Additional functionality can be added to individual GameObjects to make them behave in a certain way, e.g., a crane might need a frame loop that will allow it to update its position per rendered frame or a UI button might require click event call-backs to enable a particular operation. Such extra functionality is achieved by attaching scripts (or other components, such as rendering-related components) to the GameObjects.

2.1.2 Target platforms

Unity supports many target platforms for the applications developed around/based on it; depending on the selection of a target, a distributable, executable file can be created. The following target platforms are currently supported:

- Desktop [Windows, Linux, MacOS]
- Mobile [Android, iOS]
- Extended Reality [ARKit, ARCore, Microsoft HoloLens, Windows Mixed Reality, Magic Leap (Lumin), Oculus, PlayStation VR]
- Consoles [Xbox One/X|S, PlayStation PS4/PS5, Nintendo Switch, Google Stadia]
- WebGL

Of special interest with regards to the COGITO project is the WebGL target, allowing an application to be served via any modern web browser without the need for local installation on the user's device, regardless it being a desktop computer/laptop, a tablet or any other smart device. To this end, WebGL has been selected as the target to be used in DCC.

2.2 Overall Architecture of DCC

The application's interactions with the DTP and the IFCtoMesh Generator service are accomplished via the COGITO Unity Library as shown in Figure 2; in this sense, the COGITO Unity Library acts as the foundation upon which DCC is built. More particularly, the functionality offered by the COGITO Unity Library to DCC includes user authentication, project data retrieval (4D BIM data) as well as real-time/annotated data fetching, such as Work Progress, Health & Safety, Quality Control and IoT generated by the corresponding COGITO components. A description of the IFCtoMesh Generator service and the COGITO Unity Library is provided in deliverable D7.6. The COGITO Unity Library is distributed as a Unity Package ready to be imported in any Unity project requiring such functionality, in this case the DCC. After some initial configuration steps described in the relevant sections of deliverable D7.6, all its namespaces are available for use by the developer.

In the following subsections, the basic architectural components comprising the DCC application are described in detail; a comprehensive presentation of the tool inter-relationships can be found in deliverables D2.4 and D2.5.

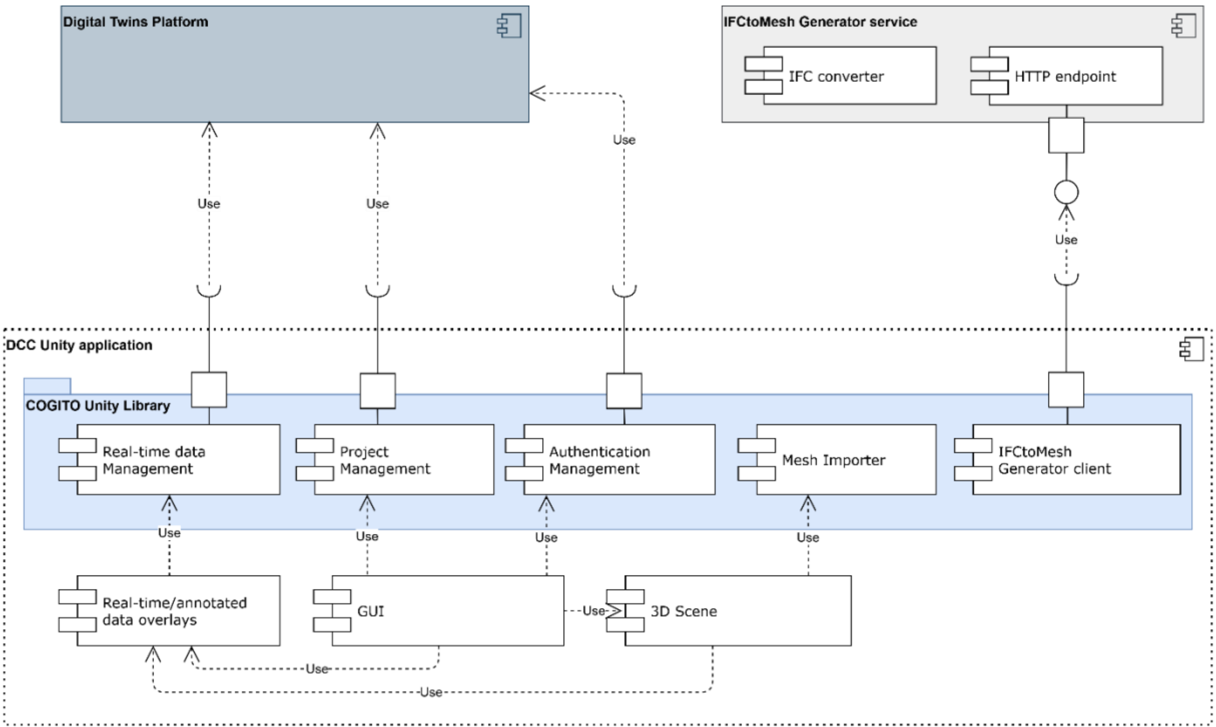


Figure 2 – High-level overview of the DCC architecture and its related subcomponents

2.2.1 3D Scene Generation

The COGITO Unity Library takes care of importing all BIM-related data into the current Unity scene; it enables the use of the IFCtoMesh Generator service, a helper service that takes care of converting the IFC files to geometry, metadata, and material data to be imported in a Unity scene. The generation of the 3D scene based on the imported geometry is handled by the built-in Unity engine.

Behind the scenes, the library imports the IFC file, converts it into geometry, metadata, and materials, adjusts and corrects the hierarchy and loads the generated GameObject hierarchy into the Unity scene (Figure 3). To provide the developer with all the required BIM information, the GameObjects generated by the library have extra scripts attached to them; the latter can convey further metadata, such as element property sets and attributes, as well as as-planned data as can be seen in Figure 4. All the included data can be used in any way the developer sees fit, for example for filtering, searching, special visualisations, etc.

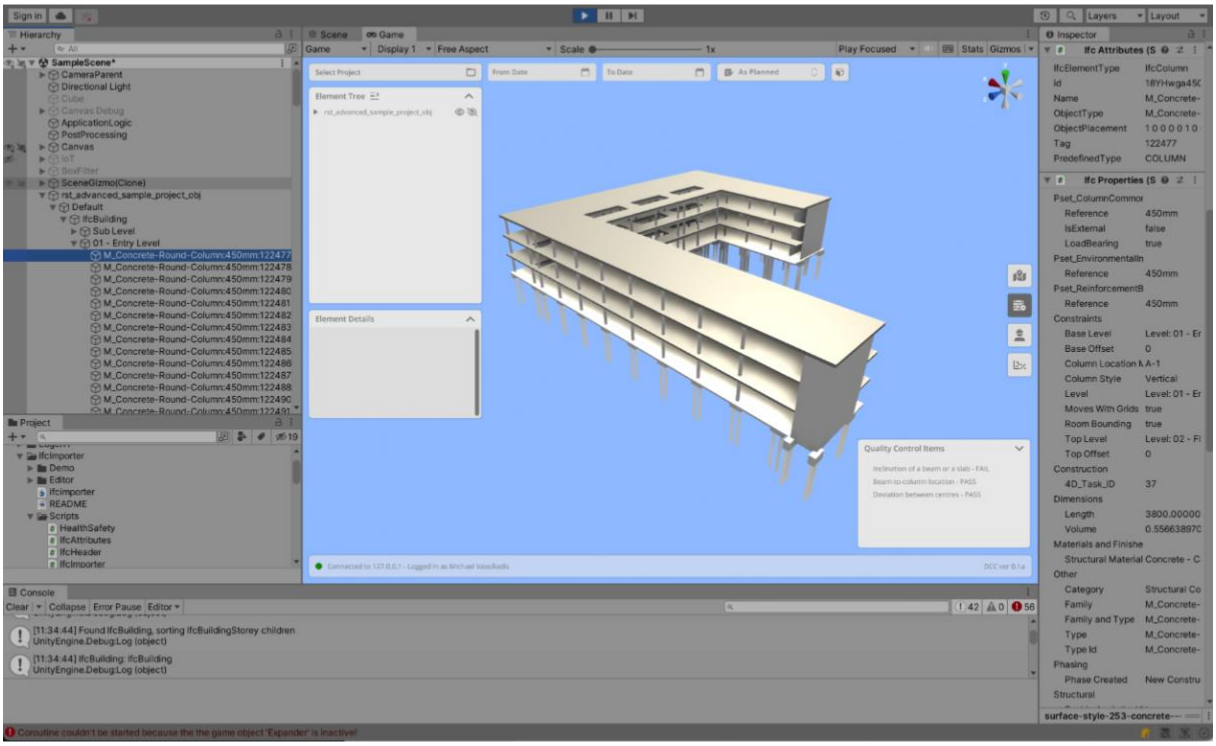


Figure 3 - Unity scene with loaded BIM

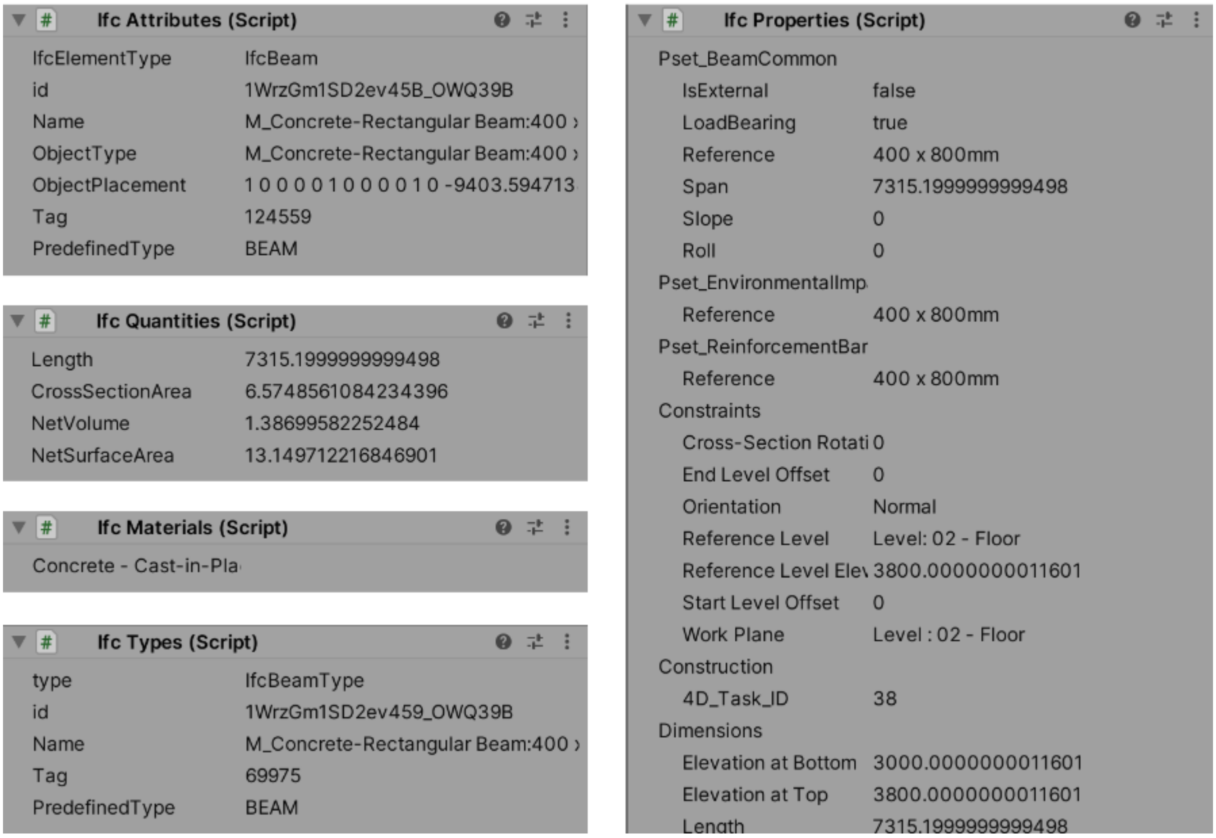


Figure 4 - Scripts with metadata attached to an IFC element

2.2.2 Graphical User Interface (GUI) Development

The application's user interface is built on top of the legacy Unity UI system. Although Unity provides a new UI framework based on web technologies called "UI Toolkit", the legacy UI system is still supported – and will remain so for the foreseeable future. The popularity of the legacy system guarantees better support from the community as well as a multitude of resources and available components greatly accelerating development.; as such, it is selected over the new UI framework. The "Modern UI Pack" from the Unity Asset Store was used as a starting point; it provides a collection of configurable UI controls, such as modal windows, buttons, dropdown menus, panels, etc. The components from the "Modern UI Pack" have been modified to suit the DCC application design; additionally, new components have been developed as needed.

2.2.3 Real-time/Annotated Data Overlays

The Overlay system depends on the Real-Time Data Manager component provided by the COGITO Unity Library. By requesting data using the `FetchOnce` or `Poll` methods (described in detail in deliverable D7.6), the DCC application renders the results either as a separate layer of objects within the 3D scene or by applying a special visual effect to the elements that are referenced by the returned data. There are three different visualisation types:

a) Extra elements/coloured spaces: Used for Health & Safety
b) 3D markers: Used for IoT/Location Tracking data
c) BIM Element colouring: Used for Work Progress and Quality Control data

The overlays can be enabled or disabled at any time using the UI (overlays button panel on the right, as described in 2.7.8); only one overlay can be active at any time.

2.3 Technology Stack and Implementation Tools

The technologies and libraries the DCC application is built upon are listed in Table 2.

Table 2 – Libraries and Technologies used in COGITO Unity Package

Library/Technology Name	Version	License
COGITO Unity Library	0.1a	Proprietary
Unity	2021.2.18f1	Proprietary
WebGL	2.0	N/A

2.4 Input, Output and API Documentation

DCC retrieves the content to be displayed utilising the COGITO Unity Library as documented in deliverable D7.6. It does not interact with any other subcomponents or tools directly; the interactions with the end-user are made solely by means of a web GUI (no API is involved).

2.5 Installation Instructions

DCC is offered in the form of Software as a Service providing a web GUI; it is deployed and maintained by its creators and as such, no installation instructions are provided.

2.6 Licensing

The Construction Digital Twin 3D Visualisation module, also referred to as the "Digital Command Centre" or DCC is licensed as a closed-source software product.

2.7 Usage Walkthrough

In short, the following steps/interactions take place upon accessing the module (also described in D2.4 & D2.5 as part of UC4.1):

1) The PM/SM/HSEM/QM signs-in to the DCC application and selects the project they would like to monitor/visualise.
2) Data is queried and retrieved from the Digital Twin Platform.
3) The user can navigate the rendered 3D BIM model. They can also activate the overlays to display further information and meta-data.
4) Having activated one of the relevant overlays, the user can click on specific elements of the 3D BIM model (or IoT devices) represented by nodes, e.g., to visualise the progress on these elements' construction, reported defects or geometric deviations, HSE issues or access workers'/heavy machinery' tracking data.

Each of the aforementioned interactions is described in the following subsections using actual GUI examples.

2.7.1 Authentication

The authentication flow supported by the DTP is implemented within the COGITO Unity Library (as described in D7.5). On the UI side, i.e., in a running DCC application, a modal window is implemented, expecting the user to enter their credentials – username and password as illustrated in Figure 5.

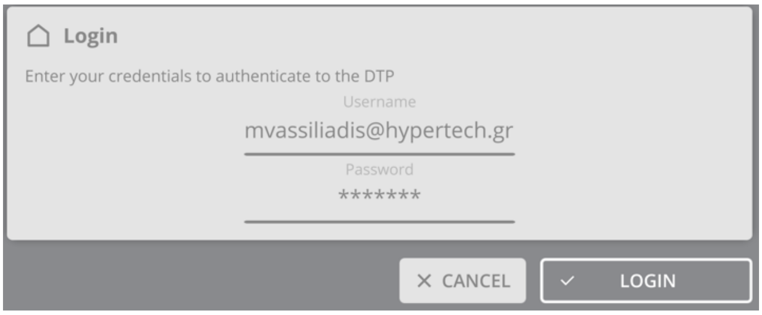


Figure 5 - Login modal window

On the backend side - prior to the application deployment - the application-specific credentials (as defined in the DTP, i.e., the Application ID and the Application Secret) need to be entered into the relevant configuration screens; the flow is executed internally and asynchronously (using a coroutine) within the library. The result is an authentication token that is stored in-memory and is subsequently used in all HTTP calls to the DTP. This configuration is performed once by the DCC developer/maintainer and does not involve any end-user interaction.

2.7.2 Project selection

Upon successful user login a valid DTP authentication token is stored in-memory – the list of projects for which the user has access rights can now be requested. This is again handled internally by the COGITO Unity Library and the results are passed to the UI layer of the DCC application for further presentation and manipulation. Clicking on the top menu button "Select Project" presents the user with a modal window containing a list of available projects as shown in Figure 6.

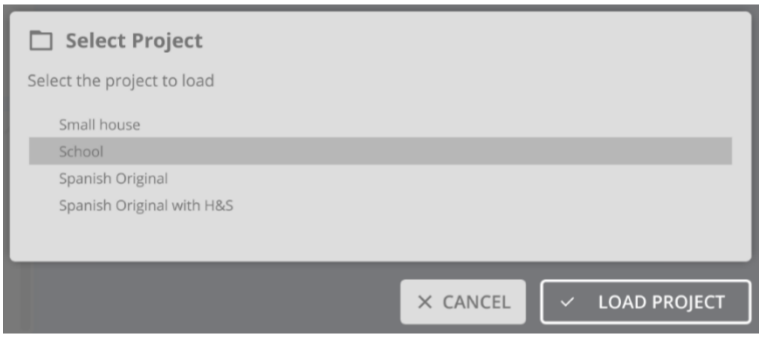


Figure 6 - Select Project modal window showing available projects

The user can select the project they would like to view and the DCC application will start executing the required conversion and importing operations to generate the required GameObject hierarchy in the scene.

2.7.3 Application Layout

Upon successful login and project selection, the user is presented with the main application layout as depicted in Figure 7. The main UI components available to the user are:

- the filtering bar (allowing the application of filtering criteria, namely time-interval selection, as-planned/as-built data, box-section mode)
- the camera widget (assisting the navigation of the user throughout the 3D model)
- the element tree listing all elements available in the BIM in two alternate hierarchies
- the element details (populated upon BIM element selection, providing all available properties and attributes)
- the overlays selection (allowing the user to visualise available metadata on top of the rendered BIM elements)
- the status bar, providing information about the connection, logged-in user and the DCC version

To avoid cluttering the UI, the element tree and element details panels can be collapsed/expanded as per user preference.

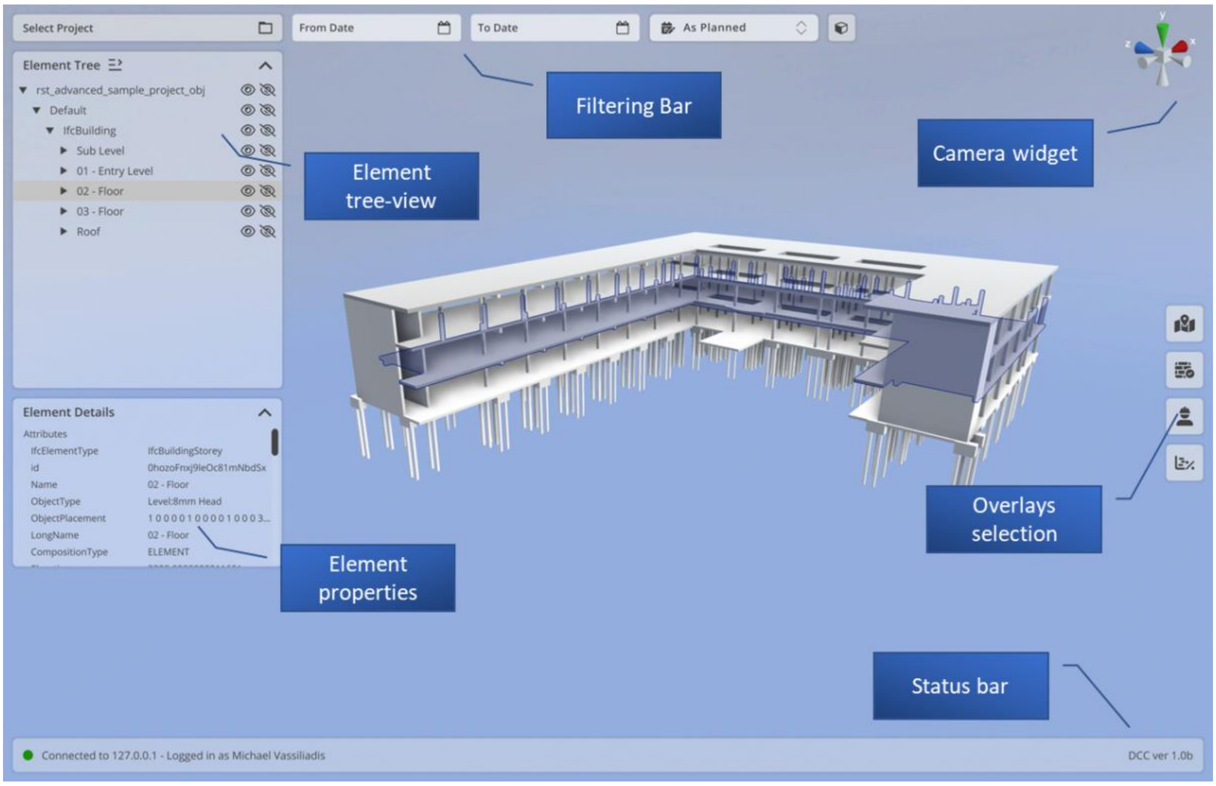


Figure 7 - Building with a selected element that contains children (floor)

As depicted in Figure 8, selecting an element in the element tree also selects it in the 3D scene and focuses the camera onto it; if the element has child elements, they are also marked as selected in the 3D scene. Equivalently, selecting an element in the 3D scene also selects it in the element tree, appropriately expanding the related hierarchy. The selected element details, i.e., its various properties and attributes are subsequently presented in the element details panel.

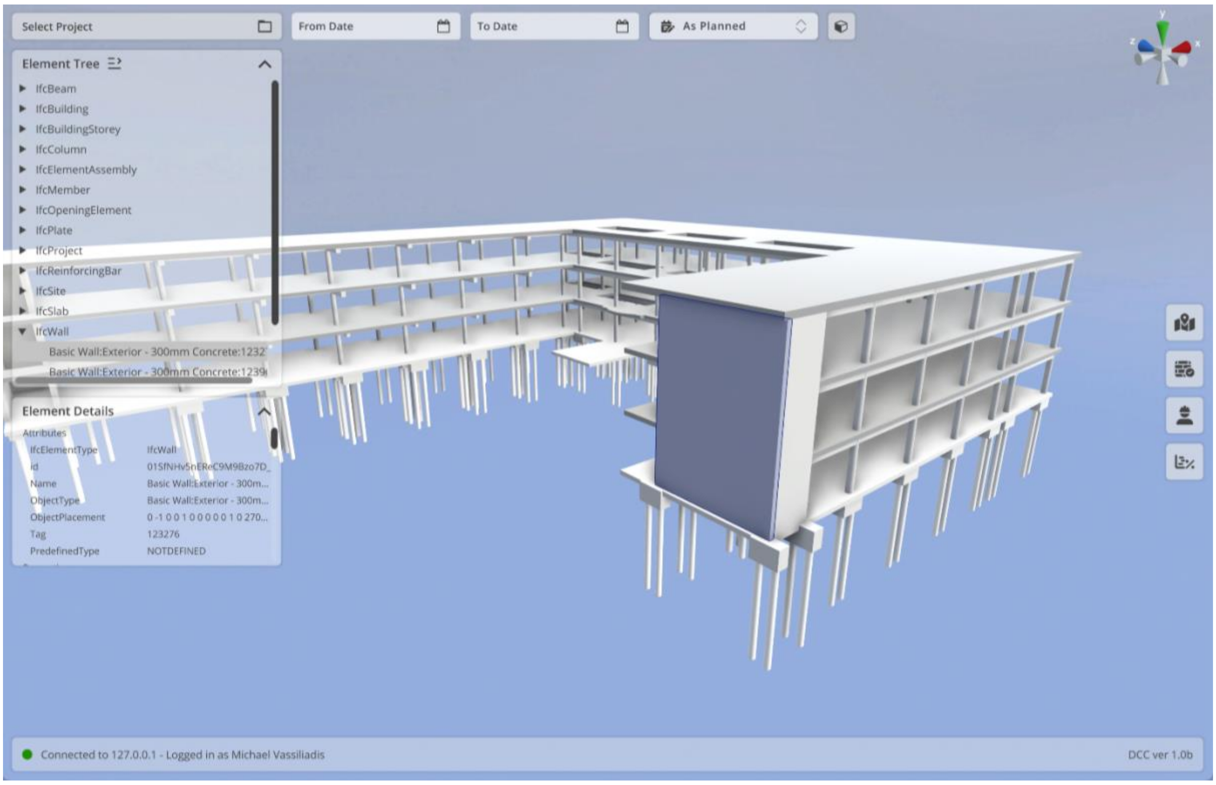


Figure 8 - Building with a selected element (wall) that has no children

2.7.4 Navigation

Navigation in a 3D scene involves the control of the camera's position and rotation using the available input method (dependent on the user device/platform used); the DCC application supports both mouse/keyboard as well as touchscreen inputs. The camera uses an orbital movement type: its rotation is fixed towards a point in the scene and the user manipulates its orbit position and distance towards that point. The user can also move that point in the 3D space allowing for panning the camera up/down and left/right. Regarding the 3D scene, the user is able to select any element by clicking on it as already explained in the previous section.

2.7.4.1 Mouse and keyboard controls

The most convenient and efficient method to control the camera is using a mouse and keyboard. Clicking and holding the right mouse button while dragging the mouse changes the orbital position of the camera, effectively revolving around the objects within the 3D scene. The mouse scroll wheel changes the distance to the fixed point (zoom-in/out) and the W, A, S, D keyboard keys move the point up, left, down and right respectively. These directions are relative to the current camera position and can translate to different axis movements depending on the current situation. The full list of mouse and keyboard controls is provided in Table 3.

Table 3 - Keyboard and mouse controls

Input	Action
Left mouse click	Selects element under cursor
Right mouse button hold and move	Orbital move of camera around objects
Mouse scroll wheel up	Move camera closer (zoom in)
Mouse scroll wheel down	Move camera away (zoom out)
Middle mouse button hold and move	Move orbit position (pan)
Keyboard W, A, S, D keys	Move the point up, left, down and right

2.7.4.2 Touch gestures

When accessed using a tablet (or other touch screen devices), DCC supports camera manipulation using touch gestures. Such devices can thus be used to navigate through a 3D scene despite lacking the convenience of a mouse and keyboard. The available touch gestures are listed in Table 4.

Table 4 - Touch-screen controls

Input	Action
Single touch	Select the element under the finger
Pinch in	Move camera away (zoom out)
Pinch out	Move camera closer (zoom in)
Hold and move finger	Orbital move of camera around objects
Two-finger hold and move	Move orbit position (pan)

2.7.4.3 Camera widget

To perform precise, single-click camera adjustment/movement such as a full top-down or side view, a camera widget is always available at the top-right corner of the screen. The camera widget represents the 3-axes defining the 3D space as well as a middle box. Clicking on any coloured axis arrow, the camera orbit is moved along this axis' edge as depicted in Figure 9.

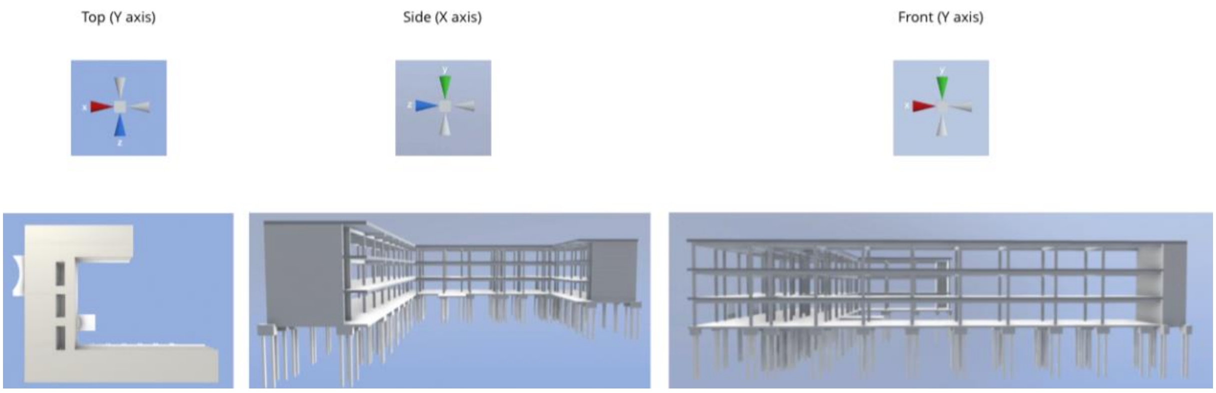


Figure 9 - Examples of precise widget camera placement

- Y moves the camera to a top-down view
- X moves the camera to a side view
- Z moves the camera to a front view

2.7.5 Element Tree panel

The Element Tree panel provides a hierarchical view of all elements in the BIM model and allows the user to show or hide the elements they choose. The panel starts fully collapsed (Figure 10) to help with the navigation in complex/large projects and can be manually expanded (Figure 11) using the expansion arrow on the right.

Figure 10: The element Tree panel fully collapsed upon DCC's initialisation

Each item in the tree view has separate show/hide controls in the form of an eye/crossed-eye button on the right. Clicking those buttons hides or shows the corresponding elements in the 3D scene. If a shown/hidden element contains child elements, then those child elements are also affected.

Figure 11 - Element tree expanded up to the floor level along with the corresponding visible show/hide buttons

Regarding the grouping of elements, the element tree provides two alternate modes of operation. The default mode groups all elements according to their position within the BIM hierarchy, as shown in Figure 11 and Figure 12. The alternate grouping mode is by element type as in Figure 13.

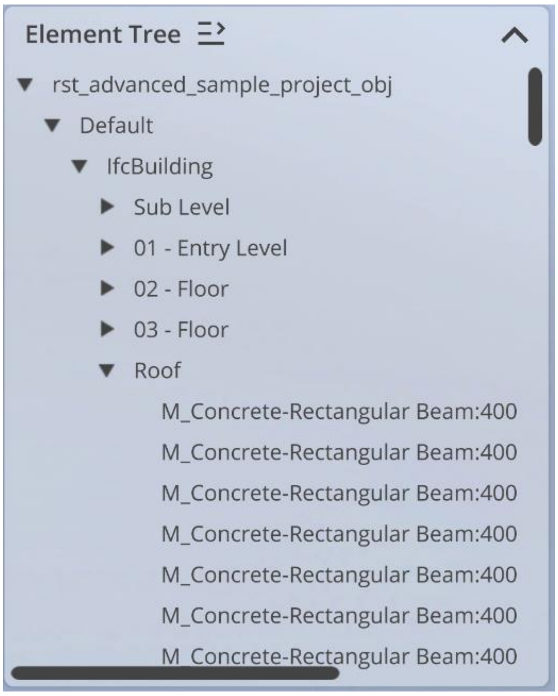


Figure 12 - Element tree expanded to the maximum depth. Left/right scrolling is available to the user

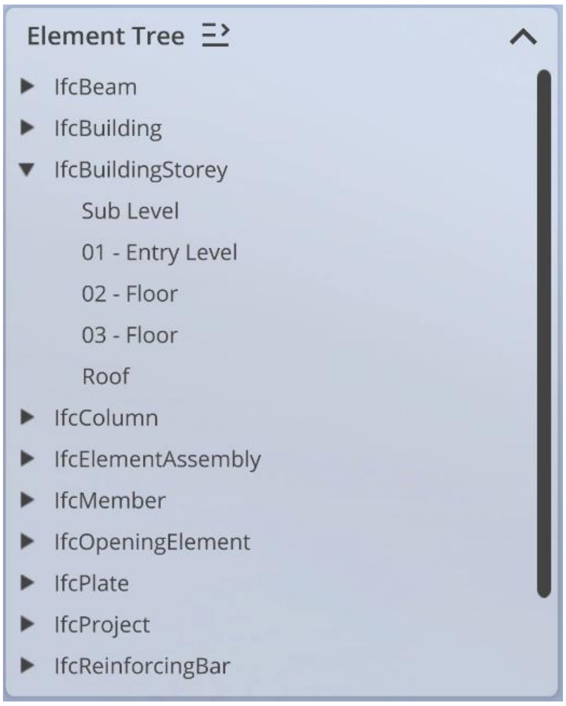


Figure 13 - Tree view in alternate grouping mode (by element type)

The user can toggle between the two modes using the header button (the icon next to the "Element Tree" heading) as shown in Figure 14. The expanding/collapsing and showing/hiding functionality remains the same for both grouping modes.

Figure 14 - Grouping mode switch button

2.7.6 Element Details panel

Selecting an element (either by clicking on an item in the tree view or using the mouse/touch gestures in the 3D scene) will populate the Element Details panel; the latter includes a property table with all available attributes and metadata linked to the particular element. An example of the Element Details panel is pictured in Figure 15. The information shown is grouped as per the following categories:

- Attributes (from the IFC file)
- Property groups (from the IFC file)
- Extra planning information (from the IFC file and accompanying data retrieved from the DTP)

Figure 15 - Populated Element Detail table

2.7.7 Filters

As illustrated in Figure 16, the application's filtering menu contains controls to allow for filtering of the data to be visualised, namely dropdown calendars for the selection of the project's time interval (from- & to-date), a drop-down menu allowing the selection of as-planned or as-built data and a button allowing to toggle between regular and box-section mode of operation.

Figure 16 - The application filters menu

2.7.7.1 Time-interval filtering

Upon project's instantiation, no time filtering is applied. The user has the option to filter the displayed elements by a date range, i.e., a start- and end-date filtering pair as depicted in Figure 17.

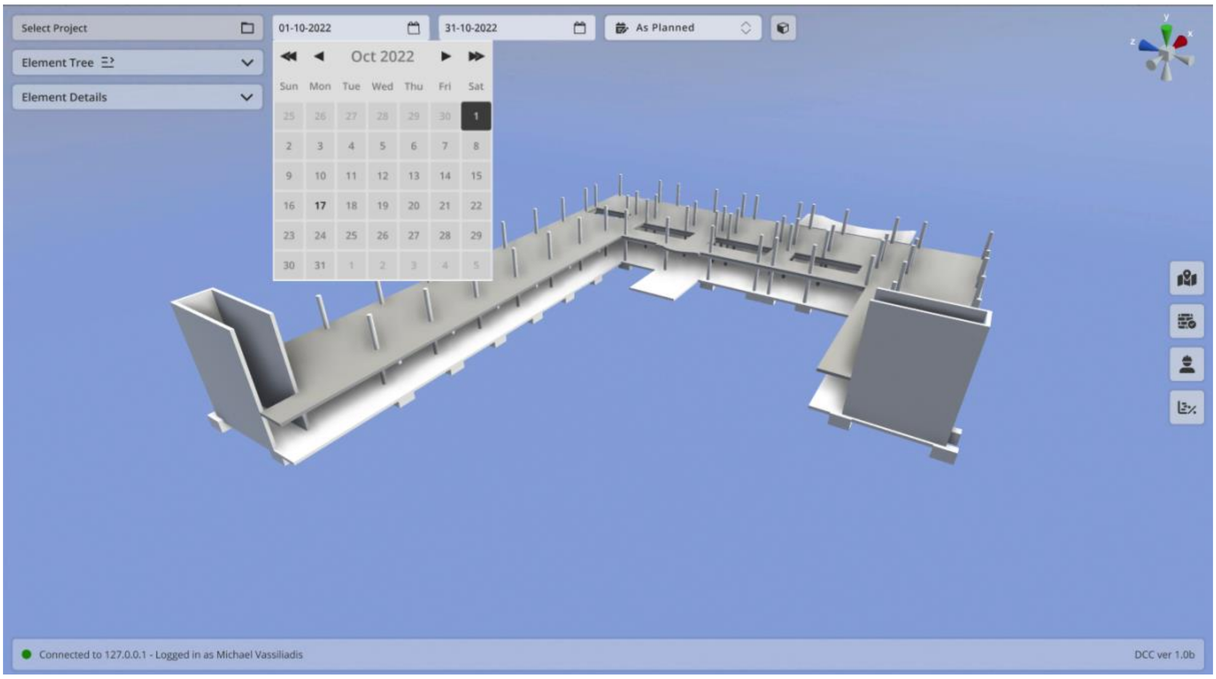


Figure 17 - Application of time-interval filtering

2.7.7.2 Schedule-based filtering

The time range taken into account during filtering can either refer to the as-planned or the as-built date of the element. The dropdown menu to the right of the date fields can be used to define the type of time range as in Figure 18:

- As Planned: Shows all elements that have their as-planned date within the selected range
- As Built: Shows all elements that have their as-built date within the selected range

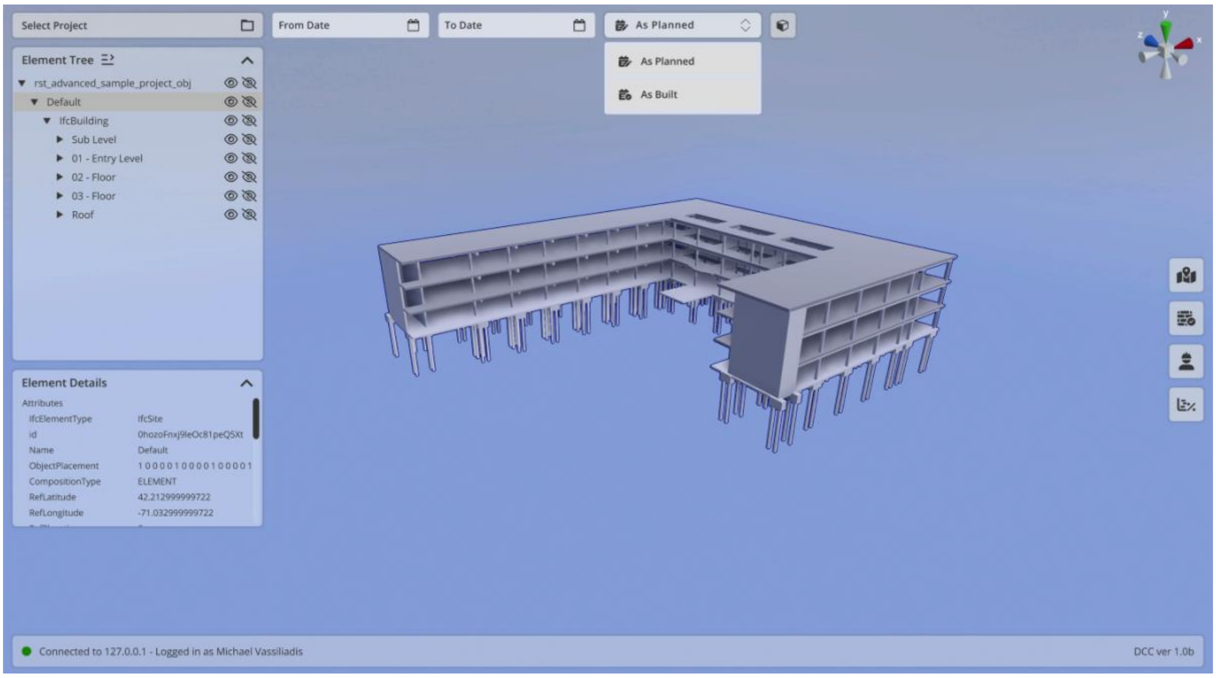


Figure 18 – Schedule-based filtering (as-planned or as-built selection for the filtered dates)

In Figure 19 and Figure 20 the two different visualisation modes, i.e., the as planned and as built modes respectively, are presented for a particular construction time interval.

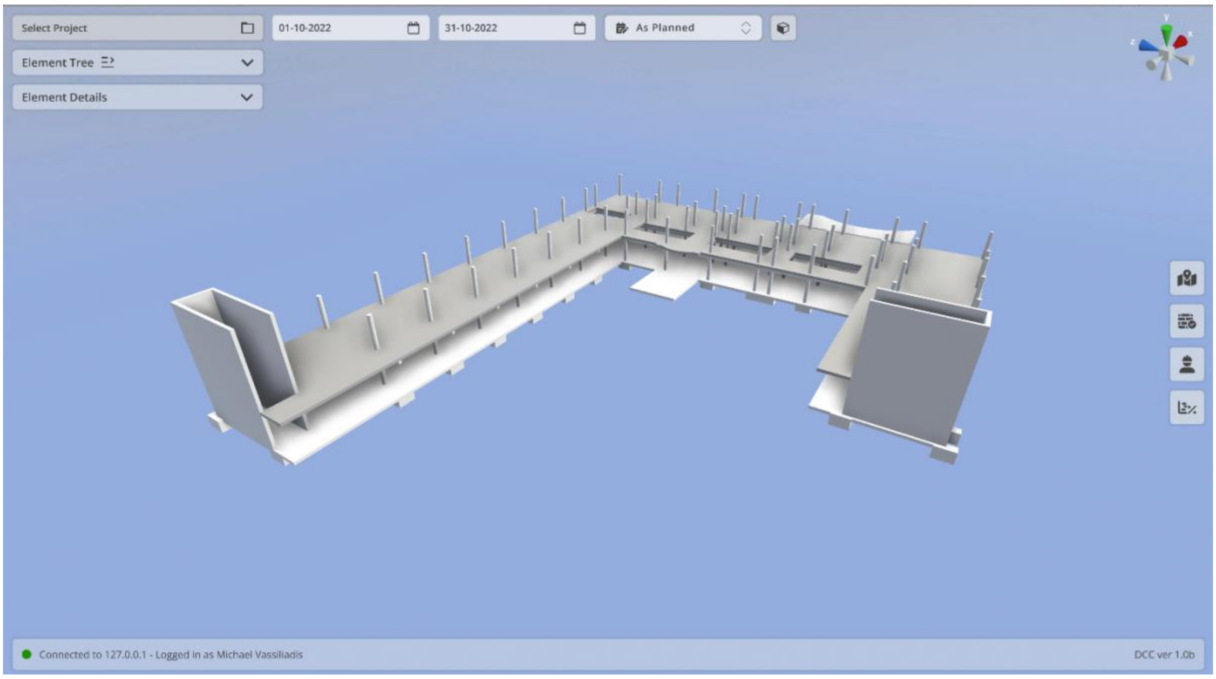


Figure 19 - Visualisation example using the as planned data as input

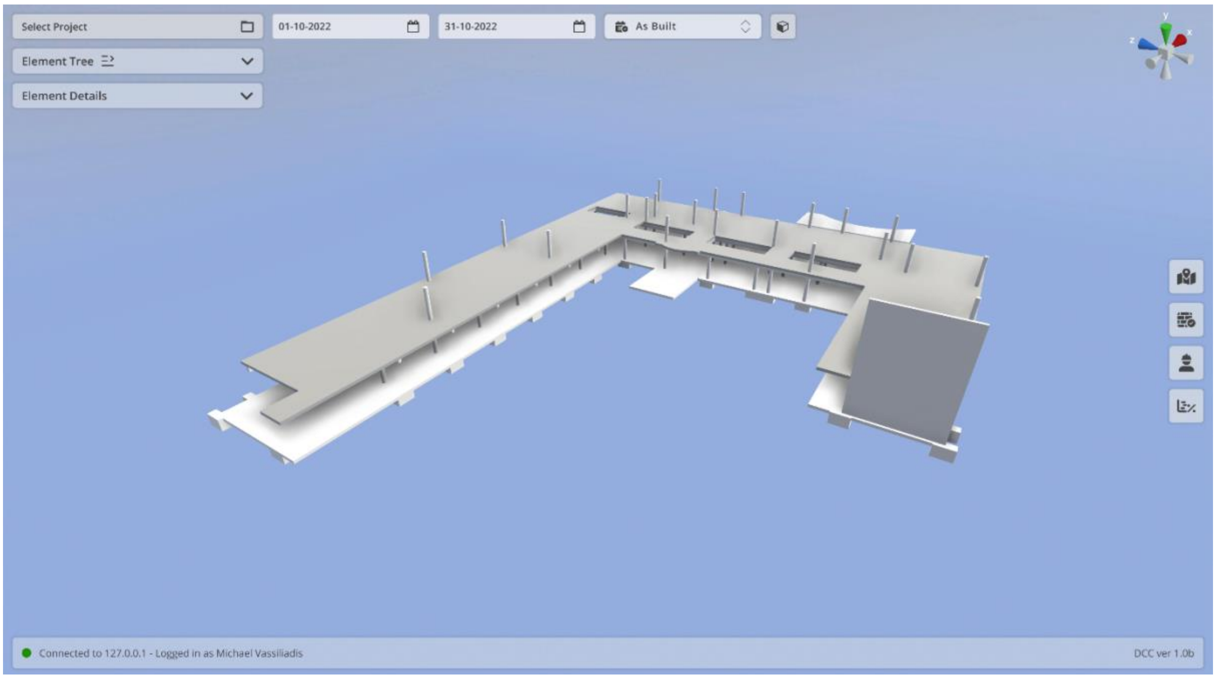


Figure 20 - Visualisation example using the as built data as input

2.7.7.3 Box filtering

The box filter allows the user to focus on a particular area of the construction. Clicking the box icon enables box filtering and presents the user with a cube and the controls to move the faces of the cube. The user can move the faces to contain the area they are interested, hiding all other elements not included in/intersecting with the box. This feature provides an alternate visualisation mode, completely uncoupled from all other application functionalities/operation modes; its development is ongoing and is expected to be delivered as part of the integration refinements taking place as part of WP8.

2.7.8 Overlay selectors

The overlay selection buttons shown in Figure 21 allow the user to enable or disable each of the four (4) available layers corresponding to the data generated by the IoT/Location Data, Quality Control, Health & Safety and Work Progress COGITO tools respectively. At any time, only a single overlay can be selected.

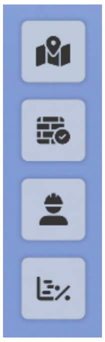


Figure 21 - Overlay button panel on the right of the screen

2.7.8.1 IoT/Location Data overlay

The IoT/Location Data overlay allows the tracking of entities capable of moving within the construction site, namely humans and machinery/heavy equipment. The available tracked entities are updated dynamically by the DTP, they are grouped by type and are displayed as 3D pins of different colour per group. An example snapshot of the DCC with this overlay enabled is provided in Figure 22.

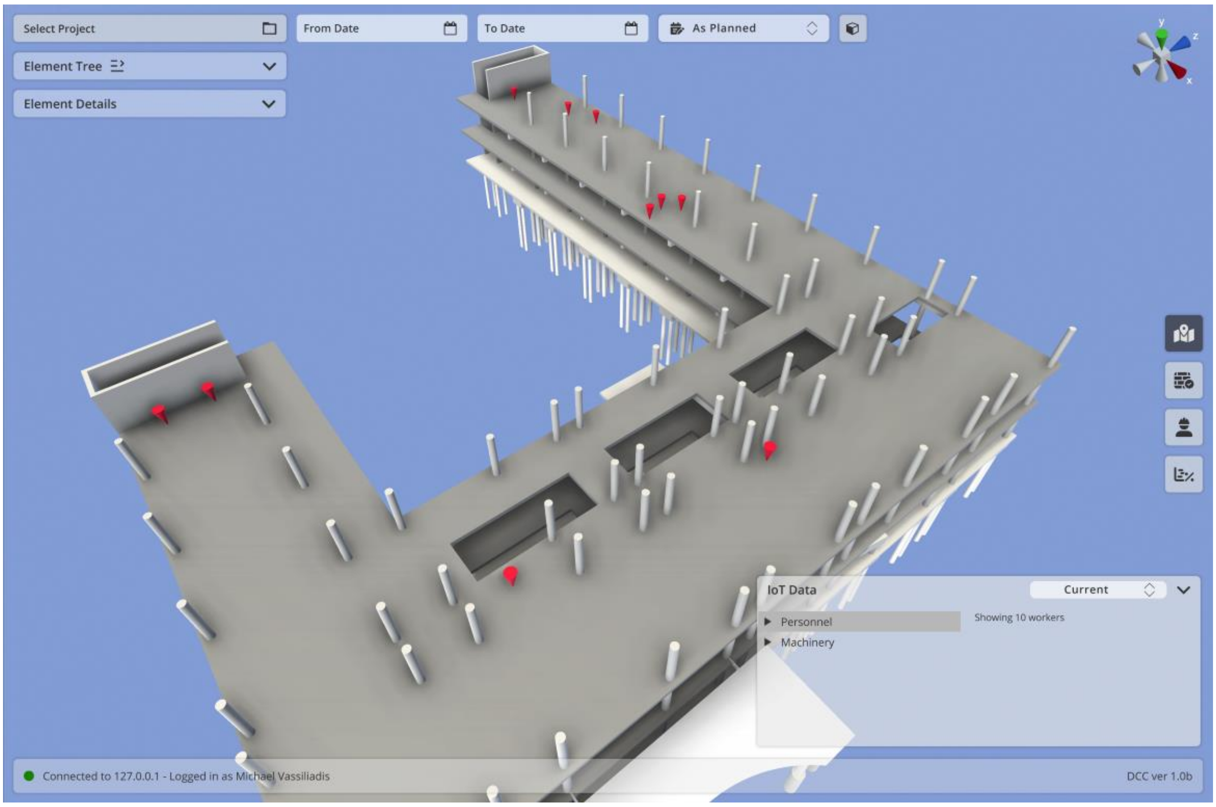


Figure 22 - Construction site visualisation with IoT/Location Data overlay enabled

In addition to the pins appearing in the 3D scene, a panel containing a list of all tracked entities grouped by type also appears at the bottom right corner of the screen. As shown in Figure 23, clicking on an item from the list selects it in the scene and provides extra information linked to its entity e.g., a worker's identifier, their group name, the tracking tag's battery level etc. A tracked entity can also be selected directly by clicking on any 3D pin in the 3D scene.

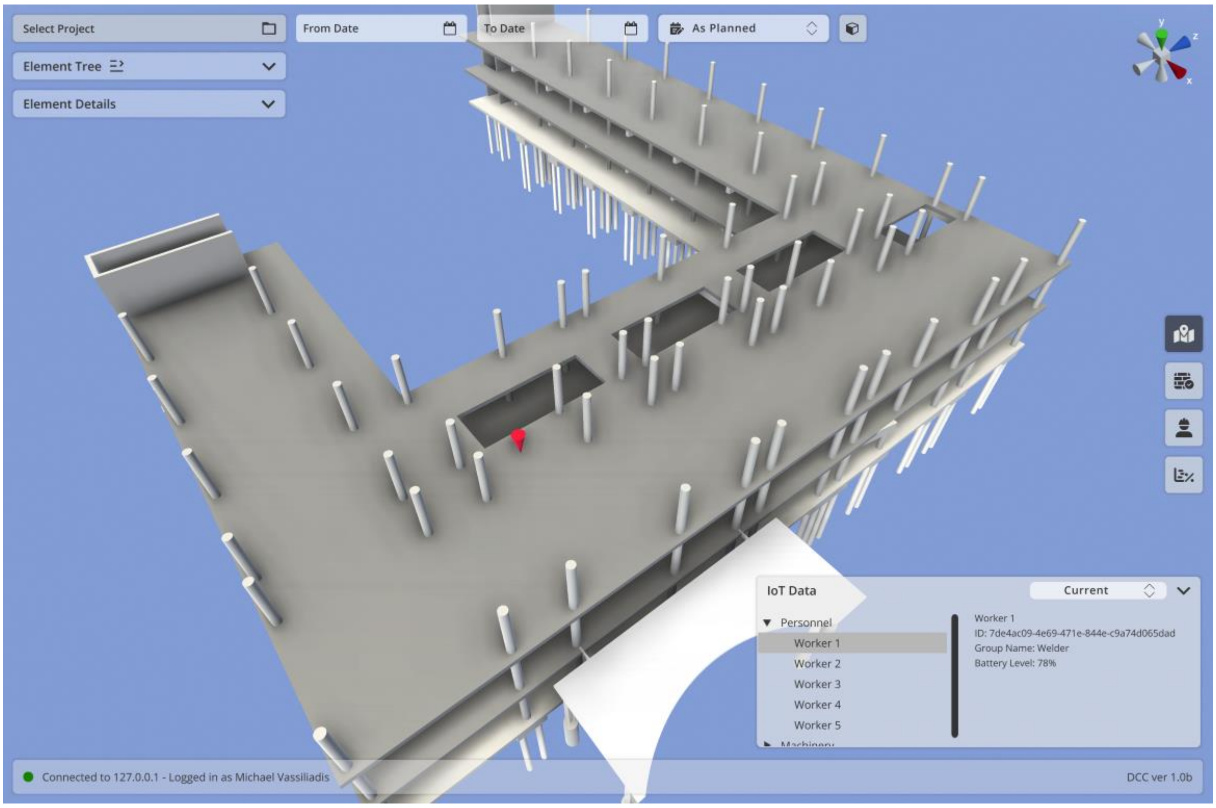


Figure 23 – Detailed information provided upon selecting a tracked entity in the IoT overlay

The overlay can either display an entity's current location at the construction site as in Figure 22 and Figure 23 or its traced path corresponding to the past 10 minutes, 1 hour or 4 hours as depicted in Figure 24.

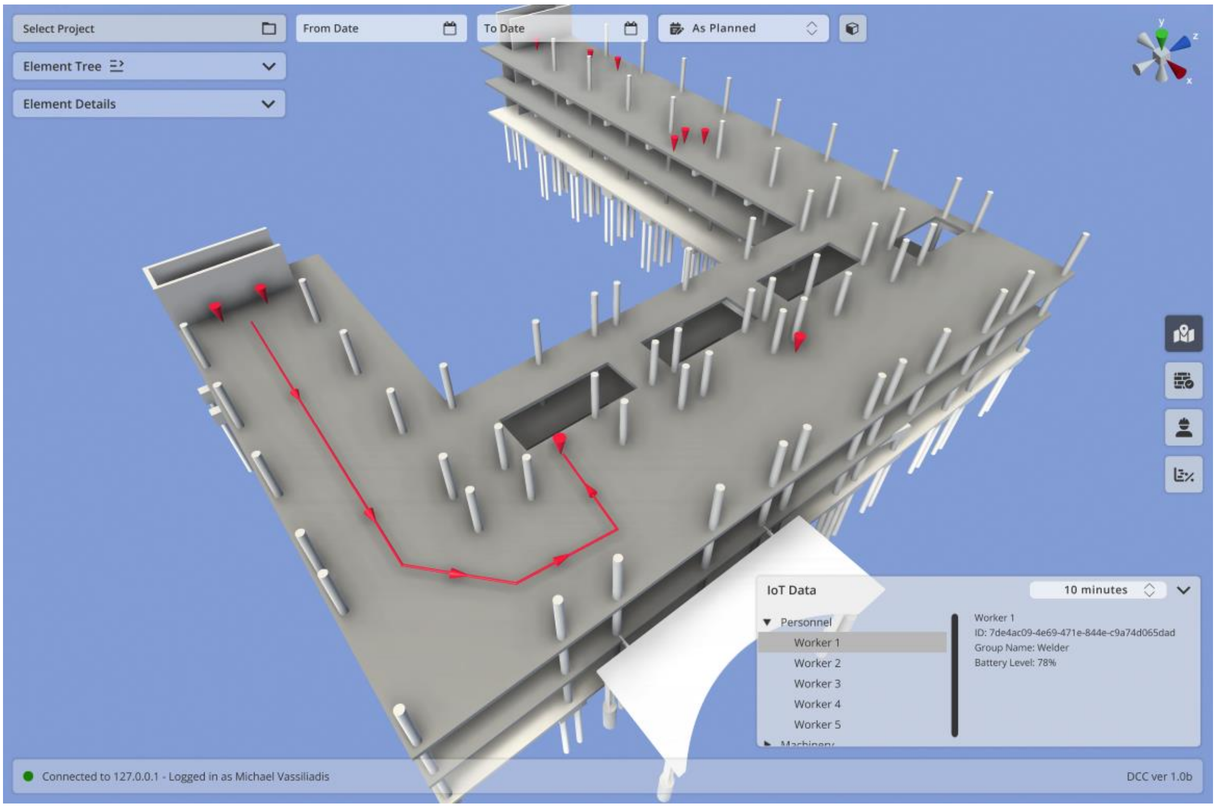


Figure 24 – Snapshot of a worker's traced path corresponding to the past 10 minutes

2.7.8.2 *Health & Safety overlay*

The Health & Safety overlay makes use of an enhanced BIM file generated by the COGITO SafeConAI component and stored in the DTP. It allows the visualisation of the various BIM elements employed to augment the construction site's safety, e.g., the installation of additional railings as depicted in Figure 25. As is the case with the IoT overlay presented above, enabling the H&S overlay also introduces a panel with all available visualisations at the bottom right of the screen. A toggle button allows the user to enable/disable time-interval filtering (if start- and end-date ranges are provided in the main filtering menu); in Figure 25 time filtering is applied while in Figure 26 it is not.

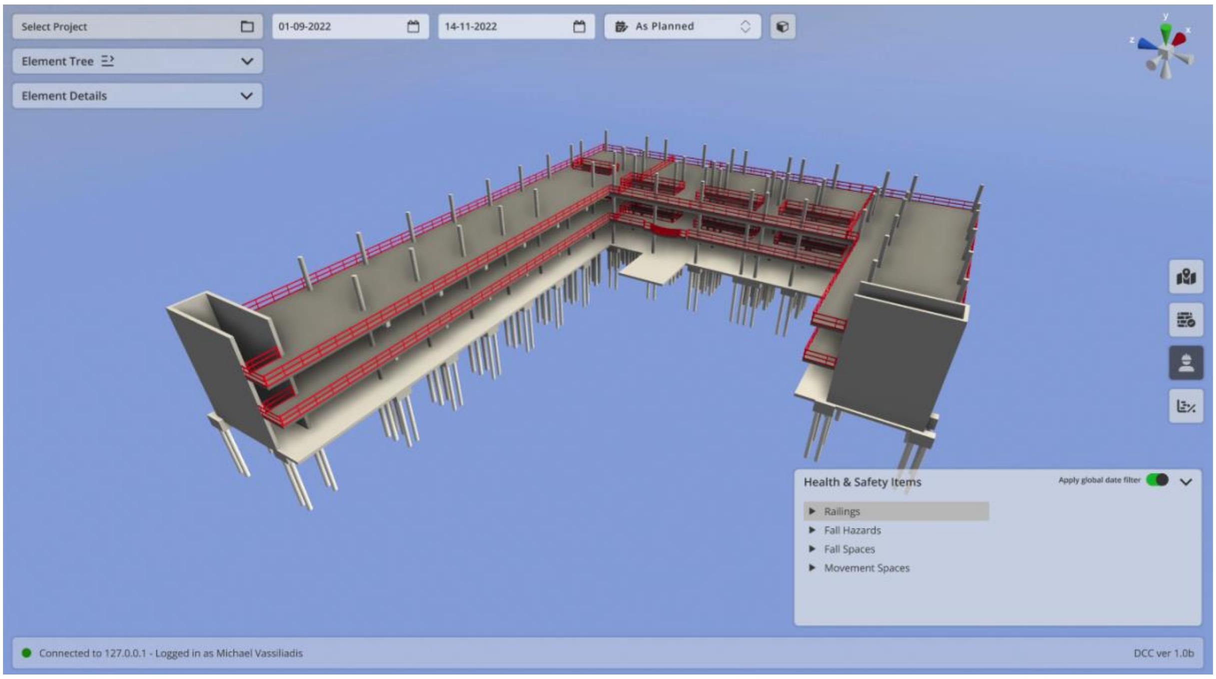


Figure 25 - Construction site with H&S overlay enabled (time-interval filtering enabled)

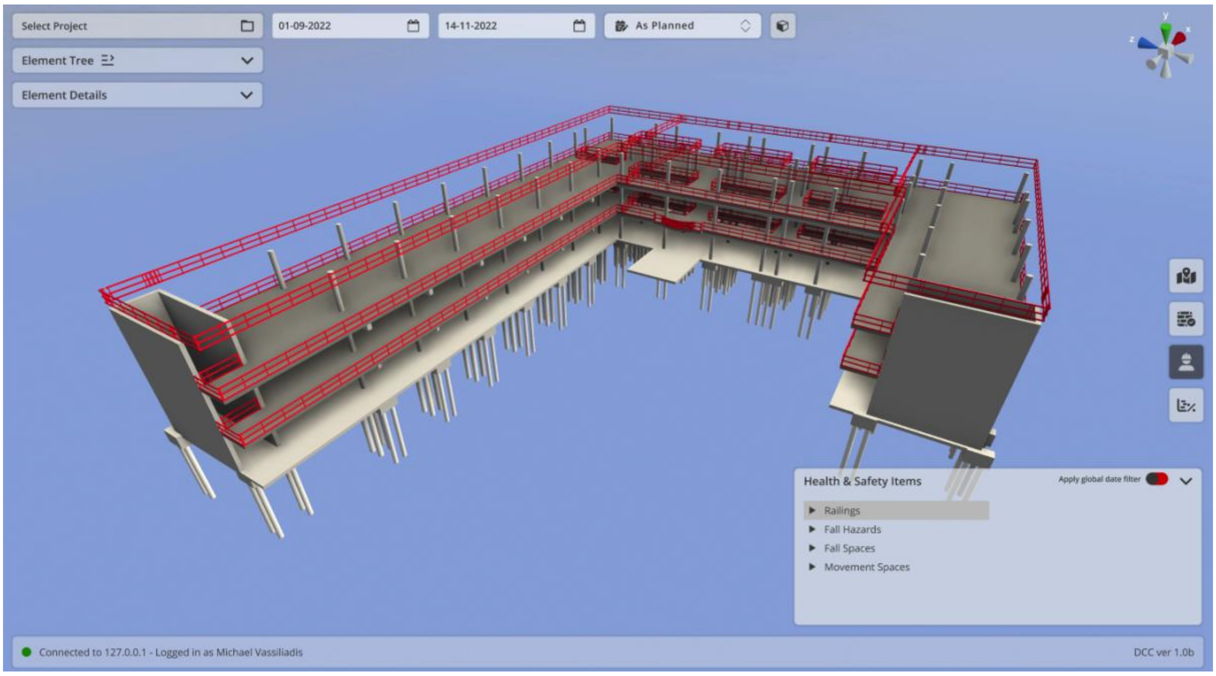


Figure 26 – Construction site with H&S overlay enabled (time-interval filtering disabled)

Selecting a particular safety element, i.e., a railing, one can retrieve information regarding its scheduled installation and removal as shown in Figure 27.

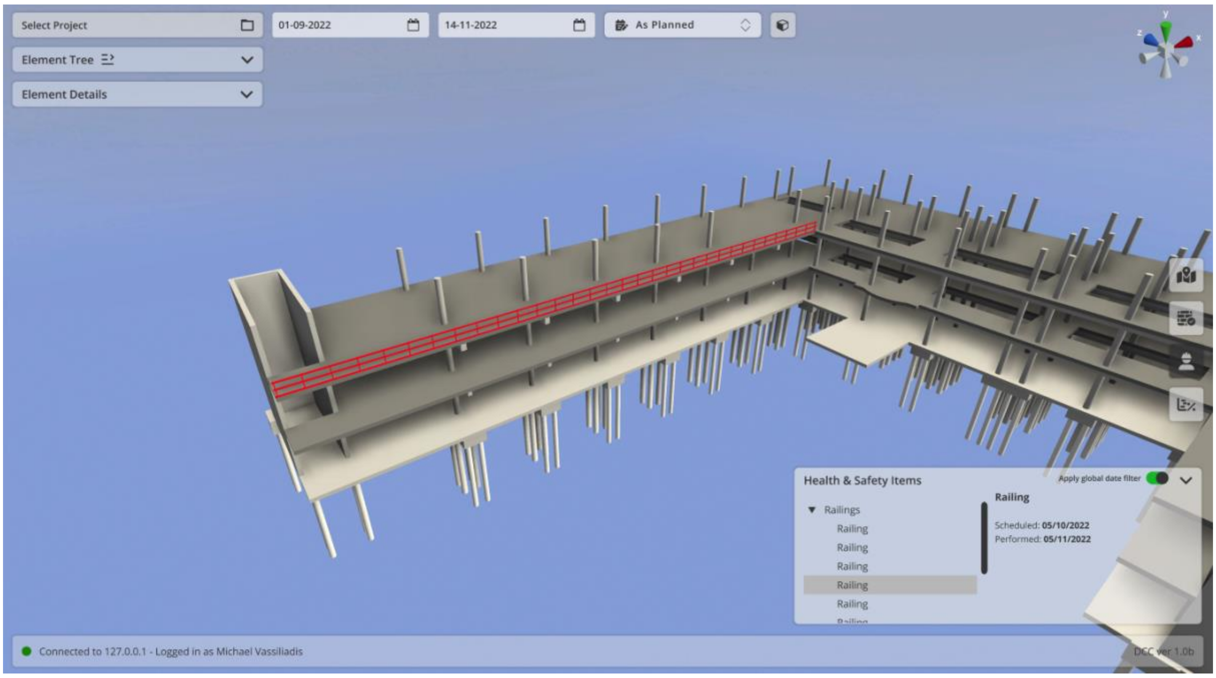


Figure 27 – Information provided for a particular safety element (railing) in the H&S overlay

Besides the required safety element-countermeasures, the user can also browse a list of the various safety-related zones and visualise them, namely the potential movement spaces (Figure 28), the fall spaces (Figure 29) and the fall hazards (Figure 30). Upon their selection, they are rendered as rectangular bounding boxes that include the area of interest.

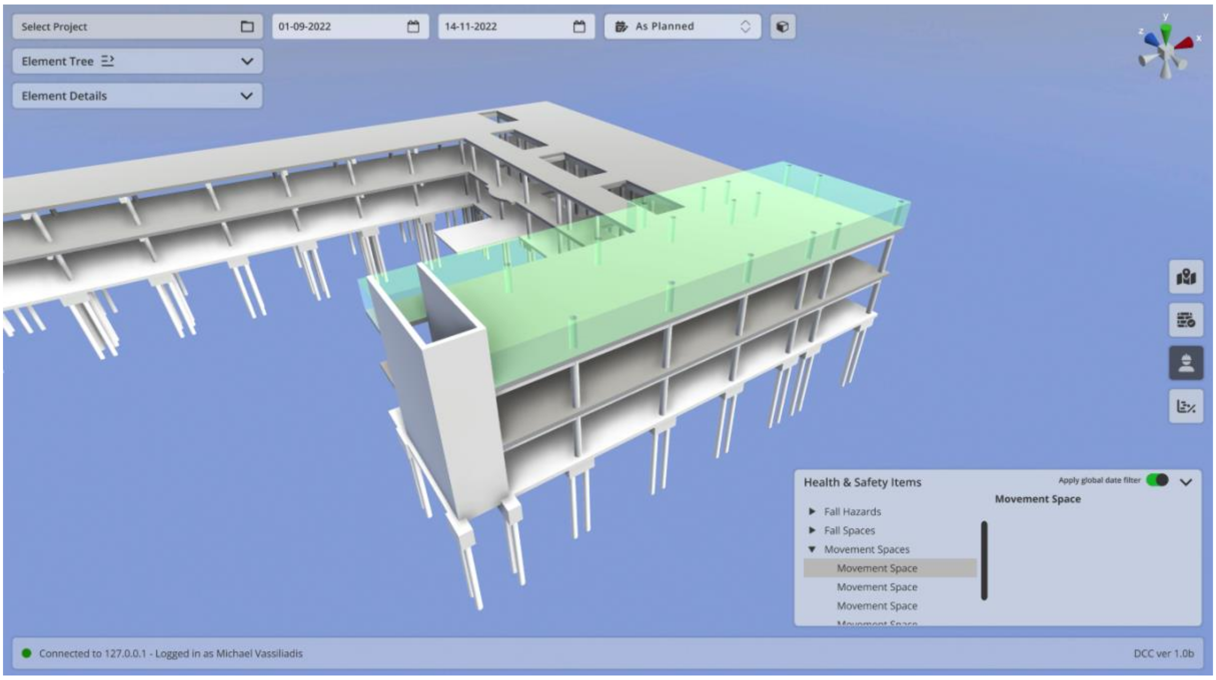


Figure 28 – Movement Space in the H&S overlay

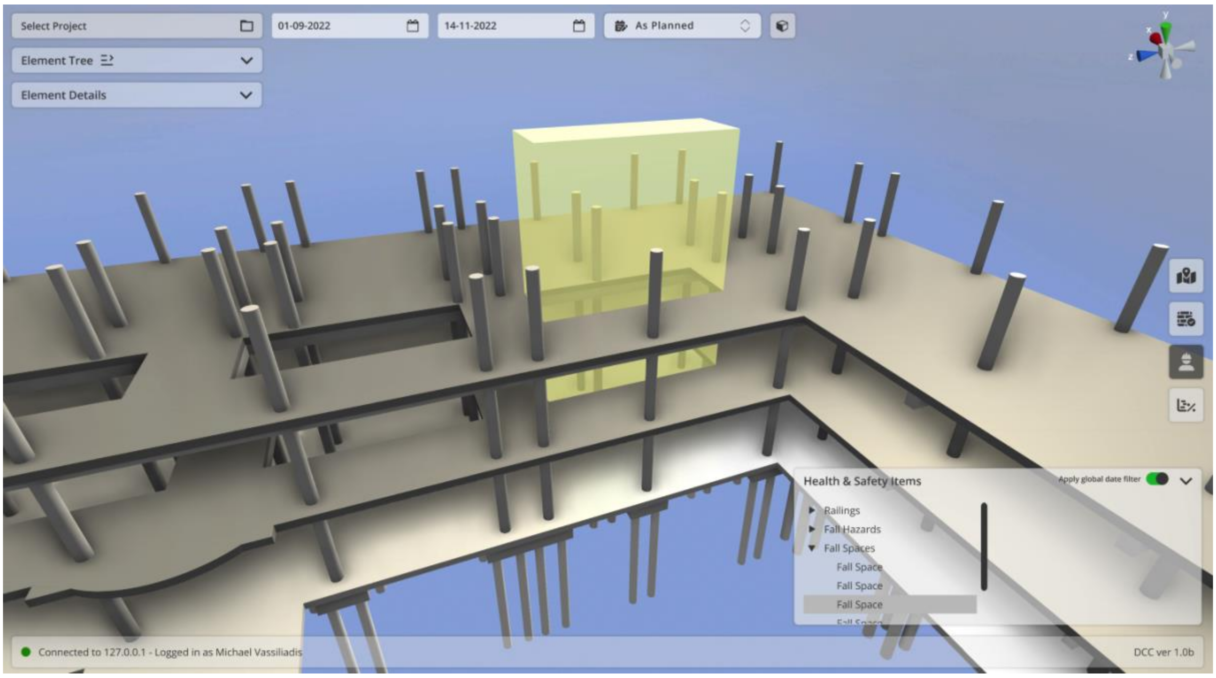


Figure 29 – Fall space in the H&S overlay

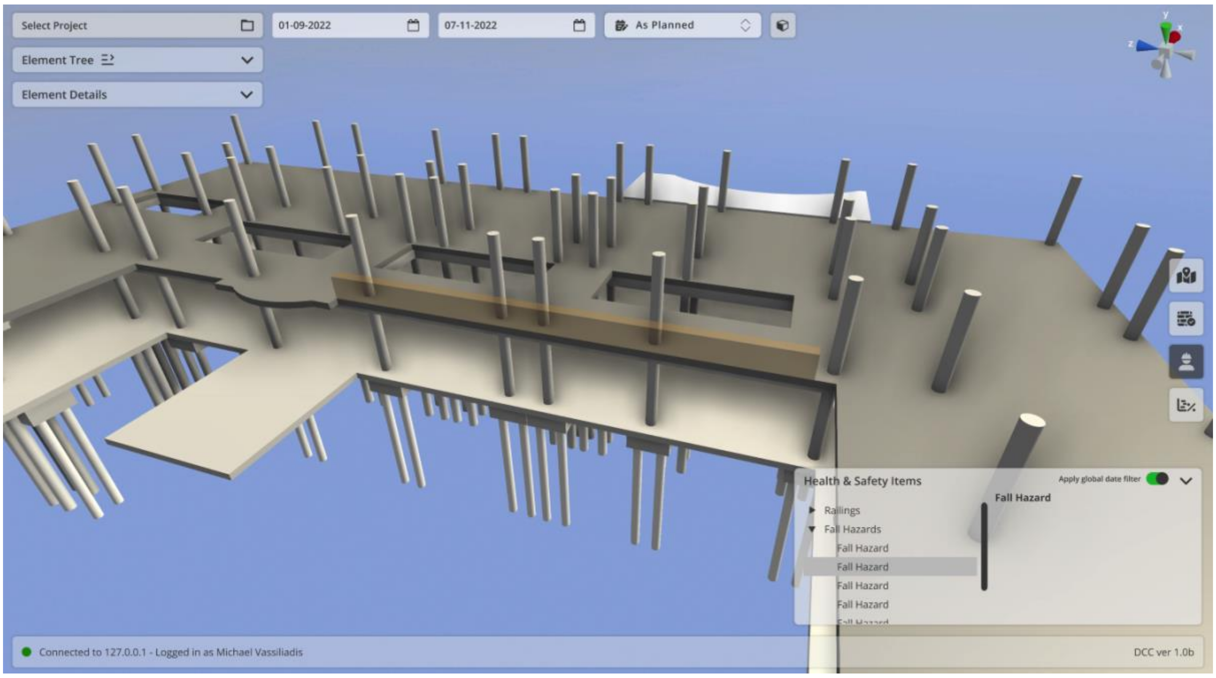


Figure 30 – Fall Hazard in the H&S overlay

2.7.8.3 Quality Control overlay

The Quality Control overlay allows the user to visualise the results obtained via COGITO's Visual and Geometric QC tools. After enabling the QC overlay the user is presented with a panel at the bottom right corner of the screen (similar to the previously mentioned overlays); the results are grouped per QC type, i.e., VisualQC or GeometricQC ones. For each QC type, there is further grouping based on the actual result (pass or fail). An example of pass and fail results for the GeometricQC case are provided in Figure 31 and Figure 32 respectively. Upon selection of a check, the corresponding elements are also marked as selected in the 3D scene (and coloured in the GQC case: green for elements passing the check, red for the failing ones). All related QC attributes/results become available to the user through the overlay's panel.

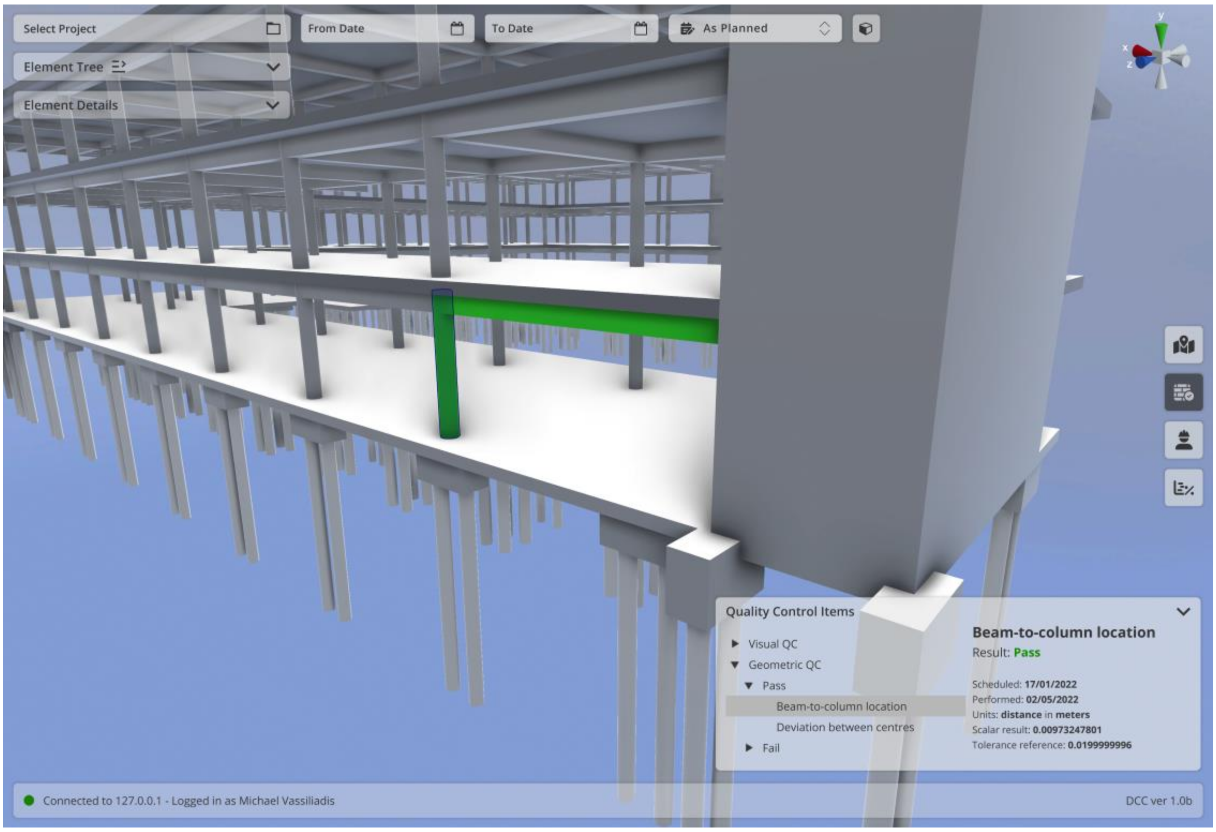


Figure 31 - Construction with the Quality Control overlay enabled (GQC case, passed check)

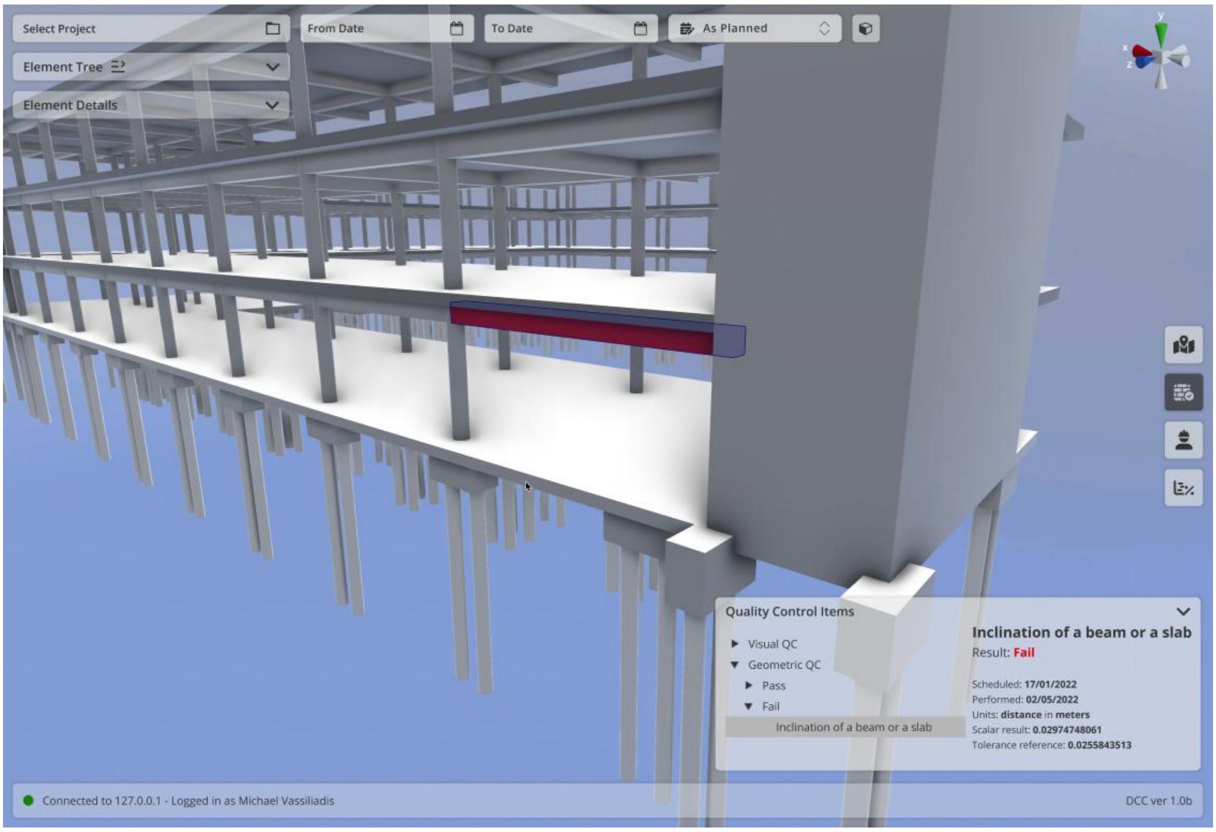


Figure 32 – Construction with the Quality Control overlay enabled (GQC case, failed check)

In the case of VisualQC results, an actual photo is also provided (if available). An example snapshot is pictured in Figure 33.

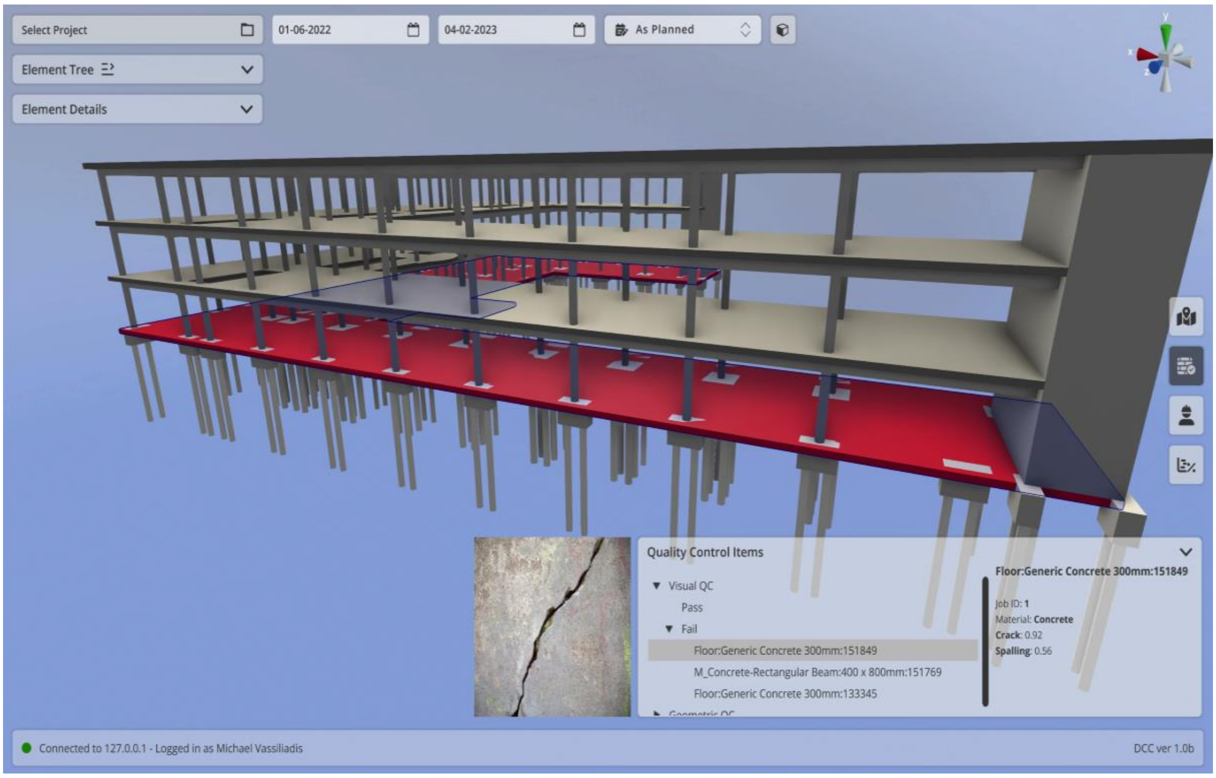


Figure 33: Construction with the Quality Control overlay enabled (VQC case, failed check)

2.7.8.4 Work Progress overlay

The Work Progress overlay provides a quick, straightforward visualisation of a project's current construction status. Upon application of the overlay, the project's BIM elements are colour-coded according to their construction progress as depicted in Figure 34.

When enabling the overlay, the DTP is queried for work progress data (originally provided to it by the WOEA/WODM COGITO components); the received data are subsequently linked to the BIM elements appearing in the 3D scene, colouring the faces of the latter using a four-colour palette to indicate the progress as per Table 5.

Table 5 – Work Progress overlay colour coding scheme

Colour	Construction Status	Completion Progress [%]
Red	Not Started	0
Yellow	Started	1-49
Orange	Progressed	50-99
Green	Finished	100

Apart from colour-coding the BIM elements, selecting a particular element either by clicking in the 3D scene or through the provided tree view at the bottom-right corner of the screen allows the user to retrieve further planning information such as the exact progress in numerical format (percentage), the as-planned start and end date as well as the actual as-built start and end date (if available). An exemplary snapshot is provided in Figure 35. The tree view provided in the overlay's panel groups elements per progress status enabling a single-click visualisation of all elements sharing the same construction status.

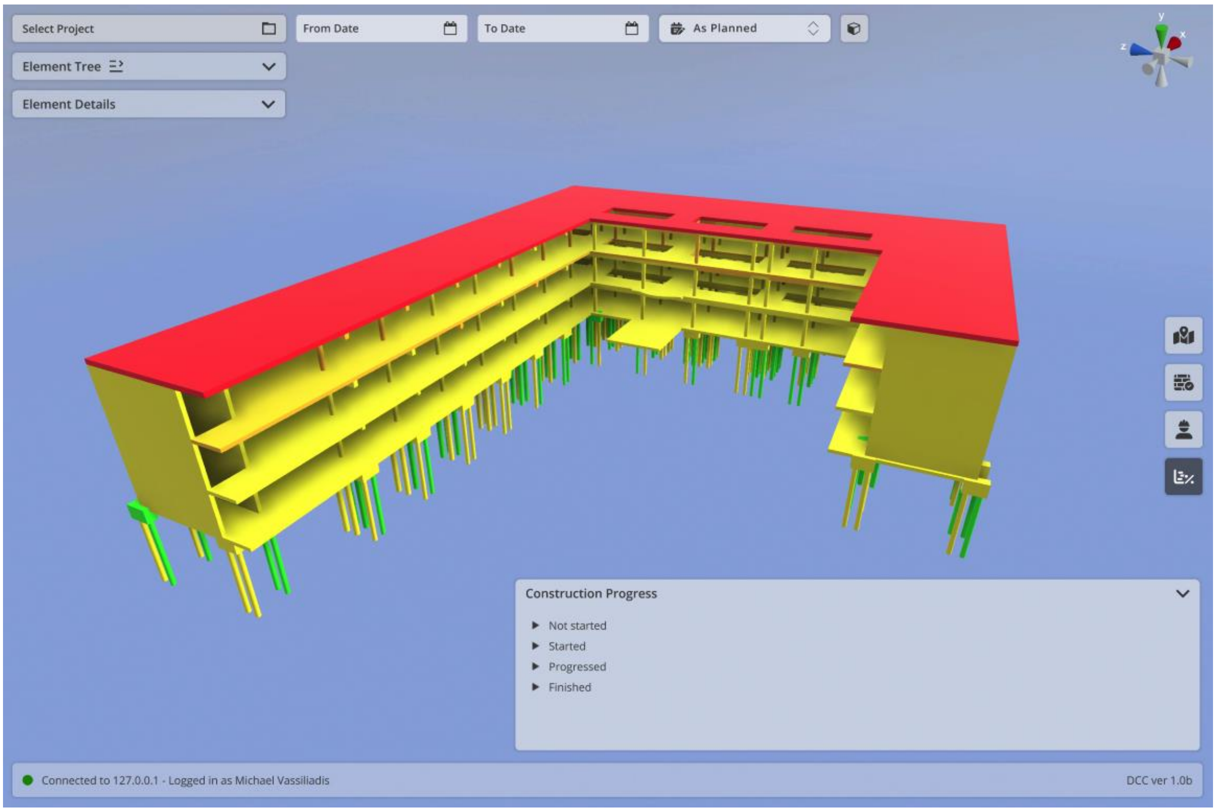


Figure 34 - Construction site visualisation after enabling the Work Progress overlay

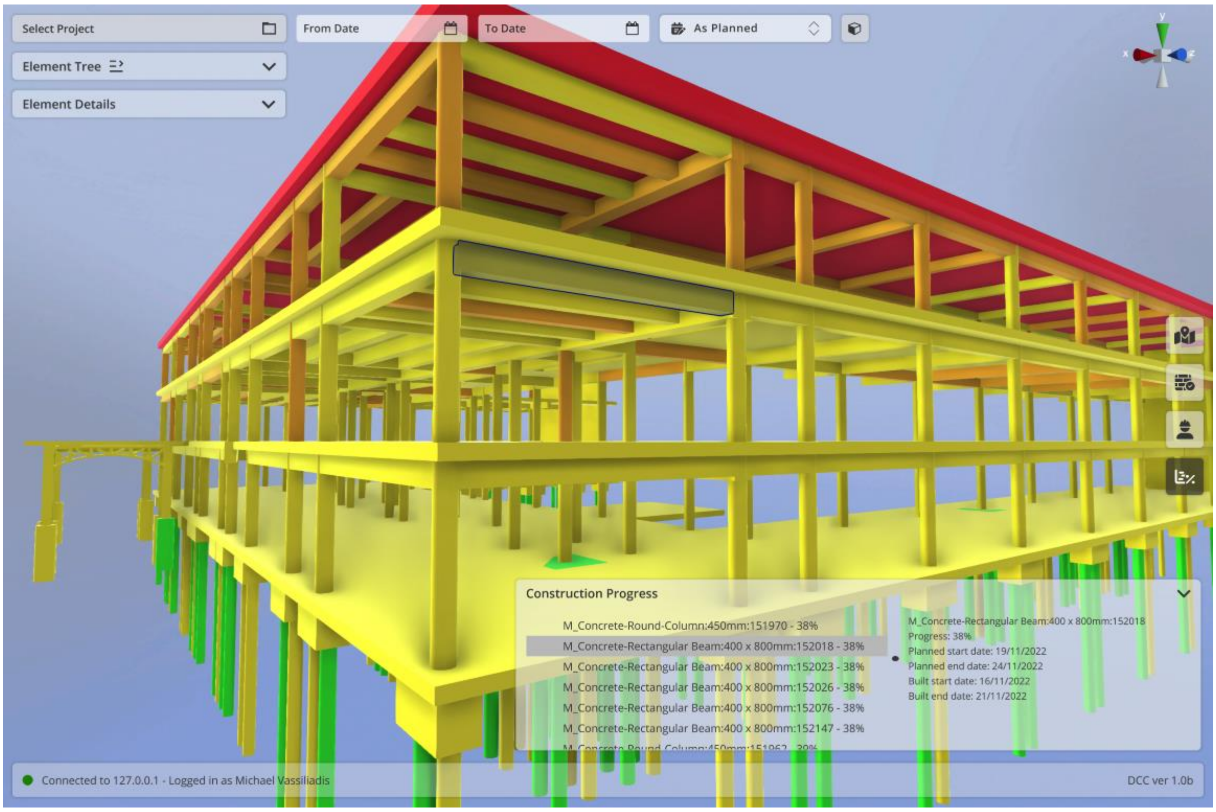


Figure 35 – Detailed information on the work progress provided upon selecting a particular BIM element in the Work Progress overlay

2.7.9 Status bar

The status bar at the bottom of the application screen contains information about the application-to-DTP connection status, the currently logged-in user (if logged-in) as well as the version of the application, as shown in Figure 36.

Figure 36 - DCC status bar

2.8 Requirements Coverage

The development of DCC (in conjunction with the COGITO Unity Library and the IFCtoMesh Generator service) is fully aligned with both the stakeholder requirements (as documented in D2.1) and the functional/non-functional requirements (as documented in D2.4 & D2.5). Any requirements/adaptations or integration work not fully addressed by the time of writing this shall be addressed in the context of WP8 and more particularly T8.1 which includes the provisions for the components' integration and fine-tuning.

In the following tables, i.e., Table 6 to Table 9, the stakeholder requirements (both in general and per BS) are listed in detail marking the status covered by the current version of the DCC.

Table 6 – Coverage of stakeholder requirements related to the computing solution

ID	Description: *The Computing solution...*	Type	Priority	Status
COGI-CS-1	[DCC, WODM, PMS, BC, SafetyConAI, VirtualSafety, gQC] Runs on desktop or laptop PC	• Operational	Must	Achieved
COGI-CS-4	runs on Windows	• Operational	Must	Achieved
COGI-CS-5	runs on Mac	• Operational	Could	Achieved
COGI-CS-6	runs on Android	• Operational	Would	Achieved
COGI-CS-7	allows access to the whole data in one location	•Operation •Functional •Design constraint	Must	Achieved
COGI-CS-8	maintains communication and data security	•Legal •Design constraint	Must	Achieved

Table 7 – Coverage of the stakeholder requirements related to the workflow, planning, execution and monitoring solution

ID	Description: *The Workflow Planning, Execution and Monitoring solution ...*	Type	Priority	
COGI-WF-6	[PMS, WODM, DCC] allows the PM and SM to efficiently detect and prioritise delays and cost escalation elements	•Functional	Should	Achieved
COGI-WF-7	[PMS, WODM, DCC] allows the PM to extract reports about: project time performance, project cost performance, costs per unit, resource consumption	•Functional	Must	Achieved (to the extent DCC is applicable)

COGI-WF-11	[WODM, WOEA, DCC] updates the activity status during work execution and monitoring	•Functional	Should	Achieved
COGI-WF-12	[WODM, DCC] allows display of activity description, planned duration and activities relationship	•Functional	Should	Achieved
COGI-WF-14	[WODM, WOEA, DCC] allows work progress reports	•Functional	Must	Achieved (to the extent DCC is applicable)
COGI-WF-15	[WODM, WOEA, DCC] updates work progress weekly	•Operational	Must	Achieved
COGI-WF-17	[DCC] uses BIM model to show tasks status information	•Functional	Should	Achieved
COGI-WF-19	[WODM, WOEA, DCC] Enables quick and easy reporting	•Performance •Design constraint	Must	Achieved
COGI-WF-25	[WODM, DCC] accesses the project execution monitoring off-site	•Functional •Design constraint	Must	Achieved

Table 8 – Coverage of the stakeholder requirements related to the Quality Control solution

ID	**Description:** ***The Quality Control (and Quality Assurance) solution …***	**Type**	**Priority**	
COGI-QC-23	[DCC, DigiTAR] allows the QM to visualise and validate automated defect detections	• Functional • Design constraint • Performance	Must	Achieved
COGI-QC-24	[DCC, DigiTAR] shows visualQC results using: graphic indicator; colourisation; and text	• Functional • Design constraint	Must	Achieved
COGI-QC-25	[DCC, DigiTAR] shows visualQC defect with contextual information (link to components, defect history, etc)	• Functional • Design constraint	Should	Achieved
COGI-QC-26	[DCC] allows defects prioritisation	• Design constraint	Should	Achieved
COGI-QC-27	[DCC] allows an advance information search/retrieval (severity, location, type, etc)	• Functional • Design constraint	Must	Achieved

Table 9 – DCC Functional & Non-Functional Requirements coverage from D2.5

ID	**Description**	**Type**	**Status**
Req-1.1	Connect and Authenticate to the DT platform (User login)	Functional	Addressed in D7.6
Req-1.2	Browse and select project from the list of available projects of the DT platform	Functional	Addressed in D7.6

Req-1.3	3D visualisation of the infrastructure's geometry (panning, rotation, camera movement and placement, walkthrough) (layer-0)	Functional	Achieved
Req-1.4	BIM-elements tree-view and element's selection (layer-0)	Functional	Achieved
Req-1.5	Resources tracking data display (layer-1)	Functional	Achieved
Req-1.6	QC defects display (layer-2)	Functional	Achieved
Req-1.7	H&S issues display (layer-3)	Functional	Achieved
Req-1.8	Tasks progress display (layer-4)	Functional	Achieved
Req-2.1	Web based App	Non-Functional	Achieved
Req-2.2	Scalability	Non-Functional	Achieved, also in tandem with D7.6
Req-2.3	Reusability	Non-Functional	Achieved, also in tandem with D7.6
Req-2.4	Interoperability	Non-Functional	Achieved, also in tandem with D7.6
Req-2.5	Security	Non-Functional	Achieved
Req-2.6	User-friendly	Non-Functional	Achieved
Req-2.7	Performance	Non-Functional	Achieved, also in tandem with D7.6

2.9 Development and integration status

DCC is still under active development; its core functionality has already been implemented and the module is expected to be complete by the next iteration of this deliverable, i.e., D7.6. It should be noted that in its current version, DCC runs as a native Unity application. Considering that it is going to be served as a web application using WebGL, upon completion of the development all the necessary optimisations must be carried-out to ensure a smooth user experience.

2.10 Assumptions and Restrictions

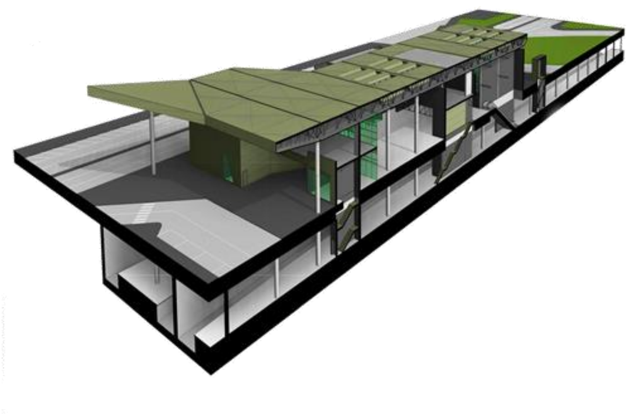
DCC is a COGITO component whose functionality is greatly dependent on data retrieved from the DTP (via the COGITO Unity Library and the IFCtoMesh Generator service); All related COGITO components, i.e., the DTP itself as well other related tools are currently pending integration. As such, the development of DCC has so far been based to great extent on mock-up BIM models and data, created in-house. The integration of tools comprising the DTP will allow a realistic testing of the features developed so far, possibly requiring adaptations or fine-tuning. The full integration of DCC with the DTP is expected to take place as part of T8.1, when all involved components should be capable of providing their full functionality and communication.

3 Conclusions

In this deliverable, the progress made towards the development and delivery of a Construction Digital Twin 3D Visualisation module is documented. The module developed, termed Digital Command Centre or DCC, is a web application enabling the PMs and other involved stakeholders to remotely access, visualise and navigate the digital twin data. DCC renders the as-planned 4D BIM model (geometry + schedule information) and as-built data, providing the option to display real-time data and annotations generated by other COGITO tools such as WODM, QC, H&S and IoT tools.

The application is developed based on the Unity game engine, utilising tools developed in T7.3, i.e., the COGITO Unity library and the IFCtoMesh Generator service described in D7.6. To provide a fully web-based experience, DCC renders the 3D geometry using WebGL allowing end-users to have access on practically any device capable of running a modern web-browser.

The development of DCC is fully aligned with all the defined end-user and functional/non-functional requirements as defined in deliverables D2.1 and D2.4/D2.5 respectively. Its current version implements all the required functionality and features and can thus be termed final; the full integration between DCC, DTP and the rest of the COGITO ecosystem is expected to take place as part of T8.1 where any required adaptations and/or required optimisations shall be addressed.



COGITO

CONSTRUCTION PHASE
DIGITAL TWIN MODEL

cogito-project.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958310