D1.4 DIGITAL TWIN INTEROPERABILITY

Timo Hartmann, TUB
Mary Lidiya Kennedy, TUB
Lucian Ungureanu,

@AshvinH2020
ASHVIN H2020 Project
www.ashvin.eu

ASHVIN has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 958161. This document reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

Project Title	Assistants for Healthy, Safe, and Productive Virtual Construction Design, Operation & Maintenance using a Digital Twin
Project Acronym	ASHVIN
Grant Agreement No	958161
Instrument	Research & Innovation Action
Topic	LC-EEB-08-2020 - Digital Building Twins
Start Date of Project	1st October 2020
Duration of Project	36 Months

Name of the deliverable	Digital Twin Interoperability
Number of the deliverable	D1.4
Related WP number and name	WP 1 – IoT driven digital twin platform
Related task number and name	T1.3 Interoperability
Deliverable dissemination level	PU
Deliverable due date	30-09-22
Deliverable submission date	30-09-22
Task leader/Main author	Timo Hartmann (TUB)
Contributing partners	CERTH and DTT
Reviewer(s)	PlanB, MFL

ABSTRACT

This deliverable presents the extract, transform, and load processes to enable information exchange. It provides a set of IDMs (Information Delivery Manuals) for the identified ETL processes and describe the reference pivot model to share data, and the operable Connectors, associated APIs, and linked data principles concerning the Ashvin tools.

KEYWORDS

Digital Twin, Building Information Modelling, Interoperability, Ashvin toolkit

REVISIONS

Version	Submission date	Comments	Author
V0.1	16.9.2022	For internal review	Ungureanu, Kennedy, Hartmann,
V1.0	30.9.2022	Incorporated comments from internal review	Hartmann

DISCLAIMER

ACRONYMS & DEFINITIONS

4D	Four-dimensional
EU	European Union
IoT	Internet of Things
API	Application Programming Interface
IDM	Information Delivery Manuals
ETL	Extract, Transform, Load
IT	Information Technology
R&I	Research and Innovation
KPI	Key Performance Indicator
CMT	Configuration Management Tool
QC	Quality Control
DES	Discrete Event Simulation
Pdf	Probability density function
HTTP	Hyper Text Transfer Protocol
CAD	Computer Assisted Drafting
VRML	Virtual Reality Modeling Language
SHM	Structural Health Monitoring

ASHVIN PROJECT

ASHVIN aims at enabling the European construction industry to significantly improve its productivity, while reducing cost and ensuring absolutely safe work conditions, by providing a proposal for a European wide digital twin standard, an open-source digital twin platform integrating IoT and image technologies, and a set of tools and demonstrated procedures to apply the platform and the standard proven to guarantee specified productivity, cost, and safety improvements. The envisioned platform will provide a digital representation of the construction product at hand and allow to collect real-time digital data before, during, and after production of the product to continuously monitor changes in the environment and within the production process. Based on the platform, ASHVIN will develop and demonstrate applications that use the digital twin data. These applications will allow it to fully leverage the potential of the IoT based digital twin platform to reach the expected impacts (better scheduling forecast by 20%; better allocation of resources and optimization of equipment usage; reduced number of accidents; reduction of construction projects). The ASHVIN solutions will overcome worker protection and privacy issues that come with the tracking of construction activities, provide means to fuse video data and sensor data, integrate geo-monitoring data, provide multi-physics simulation methods for digital representing the behavior of a product (not only its shape), provide evidence based engineering methods to design for productivity and safety, provide 4D simulation and visualization methods of construction processes, and develop a lean planning process supported by real-time data. All innovations will be demonstrated on real-world construction projects across Europe. The ASHVIN consortium combines strong R&I players from 9 EU member states with strong expertise in construction and engineering management, digital twin technology, IoT, and data security / privacy.

TABLE OF CONTENTS

INDEX OF FIGURES

1 INTRODUCTION

The main goal of the Ashvin project is to develop an integrated IoT enabled platform that can form the basis for the development of digital twins for construction sites. Additionally, the project aims at providing a set of dedicated tools to use the information provided by the platform to support design, construction, and maintenance activities. Four tools will allow managers to plan and control construction projects for increased productivity, resource efficiency, and safety. These four tools will be the main vehicle of the project to reach the envisioned impact in terms of increasing productivity, resource efficiency, and safety.

In this context, semantic interoperability is the ability of the above systems (the Ashvin platform and the Ashvin tools) to exchange information and to automatically interpret this information exchange to produce useful results (support managers). On the Ashvin project, semantic interoperability refers to the capability of the Ashvin tools and the Ashvin platform to exchange information using the same protocols.

To this end, the following closely related efforts will be employed:

1. Product testing of the Ashvin tools in relation to the platform in production scenarios on the demonstration projects. This will allow to demonstrate that the envisioned semantic interoperability can be reached beyond specified interoperability requirements.
2. Dedicated product engineering of all partners developing the Ashvin systems targeted at exchanging information that can be automatically interpreted by the different systems.
3. Industry partnerships of all organizations developing Ashvin systems.
4. The development of a common high-level ontology and API to enable this interoperability during the technical implementation of the Ashvin systems. This high-level API should form a minimal specification for the data that digital twin platforms for construction sites should make available.

This deliverable focuses on describing this API. To this end, it presents the extract, transform, and load processes (ETL) to enable information exchange. It also provides a set of IDM (Information Delivery Manuals) for the identified ETL processes of the various Ashvin systems. Together the IDMs and the API, describe the reference pivot model to share data, and the operable connectors, associated APIs, and linked data principles concerning the Ashvin tools.

It is important for the reader to realize that this report will focus on describing the interoperability requirements to enable the communication between the Ashvin systems. The deliverable will not summarize system internal exchange measures, data processing pipelines, or ontologies. These results of the Ashvin project will be reported in the respective dedicated deliverables:

- “D1.2 An ontology for digital twin models for the construction industry” that describes the ontological approaches of the Ashvin platform
- “D3.3 Data fusion methods and digital twin based IoT data processing pipelines” that describes the specific data transformation and fusing operations of the Ashvin platform

- D4.2 through D4.6 that describe the specific Ashvin tools to support construction site planning and control based on digital twin information

This report will also not focus on developing specific APIs for the interoperability requirements with respect to the Ashvin tools for supporting evidence-based design (WP2) and maintenance (WP5). The developments of WP2 will focus on demonstrating possibilities for using past digital twin information from construction sites to inform future design efforts. As little experience with historical digital twin implementation on construction projects is available, the results of WP2 will not yet allow for defining sound generalizable interoperability solutions. The deliverables in WP2 will provide first suggestions for data exchange and interoperability requirements, but these requirements cannot yet be meaningfully integrated in the API presented here.

At the same time, much work has been completed in the last years to suggest interoperability solutions to support asset and facility management. Standard APIs do already exist in this area and will therefore also not be included in the API presented here. We will provide a summary of the current state of the art in the area of maintenance in section 3 providing references to some of the most important standards existing. We propose that readers with an interest in asset management refer to these standards.

1.1 Intended audience and purpose

The report is targeted at software developers that intend to develop digital twin software systems for construction sites, be it general digital twin platforms or specific management tools to make use of digital twin data. Additionally, the report is targeted towards IT managers and R&I professionals that plan to implement specific digital twin systems and solutions on specific construction projects. While the report is first and foremost meant to structure the Ashvin system development strategies, we try to generalize the proposed interoperability concepts to also provide value to software developers and managers beyond the direct context of the Ashvin project. We believe that the interoperability approaches designed on the Ashvin projects represent best industry practices that can be used as input for future standard development efforts once digital twin technologies have matured on construction projects.

1.2 Outline

The deliverable is structured as follows. To provide the reader with the required context, the next section summarized the overall Ashvin platform architecture and the envisioned Ashvin toolkit for supporting construction site planning and control. Based on this the information requirements for each of the Ashvin systems is introduced, followed by a proposal for a common API framework to meaningfully exchange information between these systems. At the end, some recommendations for implementing the API on top of an existing digital twin platform are provided and illustrated with specifics from the Ashvin project.

1.3 The Ashvin IoT enabled digital twin platform

Figure 1 provides an overview of the architecture of the Ashvin platform that shows how information is collected from IoT devices, such as sensors and cameras, and processed within the integrated digital twin platform. The upper part of the platform is the Analytics layer that can integrate advanced AI prediction methods, video / image processing approaches, and other business intelligence algorithms. The output of these algorithms needs to be made available to a range of different tools that can tie into the platform and provide dedicated decision support to construction professionals. To cater the user and context centric design of these tools, but also their continuous and agile improvement, the platform should provide an open API that makes the results of the analytics layer available.

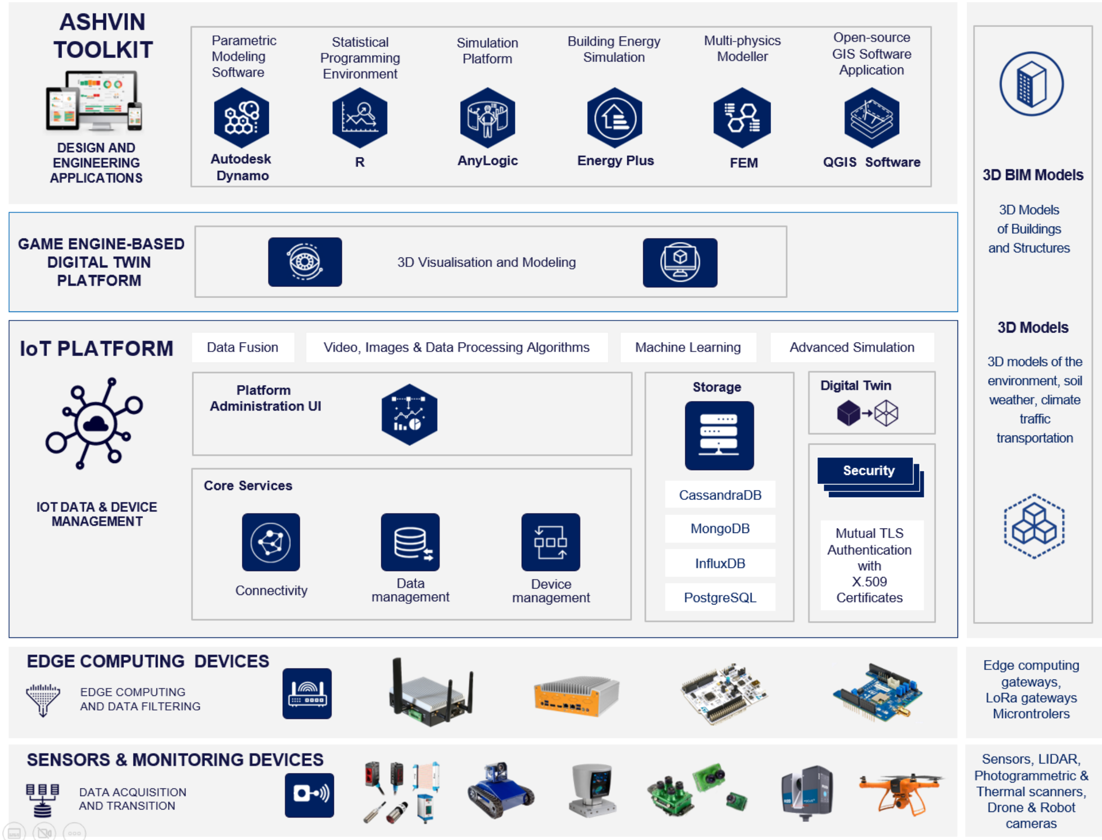

Figure 1: Overview of the Architecture of the Ashvin platform

During the Ashvin project, we developed four dedicated tools making use of the information from the Ashvin platform. These tools cover the main information requirements of construction managers in terms of

- monitoring past construction work with respect to the resources used,
- controlling whether work has been completed in sufficient quality and fits with the overall construction system that is realized,
- planning future work using past productivity information,
- and evaluating site safety.

The API presented here was designed for the data requirements of these tools. The following sub-sections briefly introduce the different tools to provide readers the practical context for the functionality of the proposed API.

1.4 Simulation-based real-time construction site and logistics planning tool (developed in T4.2)

The in T4.2 developed simulation-based planning tool facilitates improved management of upcoming construction activities by stochastic discrete event simulations based on real-time data. During execution of selected construction work process patterns real-time data regarding equipment's movements are collected. These collected data are used to control the construction process and to determine stochastic productivity rates of the different construction activities. In the following, the stochastic productivity rates can be used as input parameters for data-driven discrete event simulation within the simulation-based planning tool. By collecting continuously real-time data on site, productivity rates can be updated to provide meaningful inputs by considering real life variability within the simulation-based planning tool. Additionally, the simulation-based planning tool can consider supply-chain logistical, resource dependent, and spatial characteristics of the site. Users of the tool can test different delivery options by changing delivery times or amounts of material, resource allocations by changing the number of workers or equipment, and site lay-outs by dividing construction site in different workspaces. Due to the stochastic approach and the possible different options a multitude of future construction possibilities can be modelled within the simulation-based planning tool. The tool predicts the duration of stay for each trade at a workspace, calculates active as well as idle times, and creates an output table containing this information. Furthermore, the tool calculates the in T4.1 developed KPIs for productive, resource efficient, and safe construction works automatically and, thus, the effects of different construction possibilities can be compared. Idle times for resources, inefficient allocation of resources, and time-space conflicts of trades can be avoided. Due to the usage of the simulation-based planning tool future construction execution can be optimised according to construction relevant KPIs based on digital twin data.

1.5 Configuration management tool to track as-designed and as-built, as well as, to allow for seamless commissioning (developed in T4.5)

The configuration management tool (CMT) being developed under T4.5 aims to establish and create a quality control (QC) procedure by maintaining consistency among the requirements, design, and configured items. The tool uses digital twin data to establish the consistency of QC procedures. Real-time data gathered from the construction sites are used for the As-Planned vs As-Built comparison. Primarily this tool enables the user to establish QC, change management through the lifecycle using real-time digital twin data and the impact of the changes.

1.6 CMT - Construction monitoring tool with productivity and safety KPI decision making dashboard (developed in T4.6)

The construction monitoring tool visualizes past construction sequences using 4D visualizations. The visualizations support project managers to understand completed construction activities and space allocation so that a clear understanding of the current site conditions can be obtained. The monitoring tool also provides an overview of the actual status of a construction site using the Ashvin KPIs published within D4.1.

1.7 Privacy ensuring safety management, simulation, and training tool (developed in T4.6)

The safety management tool of the Ashvin project, is a monitoring and dashboarding tool specifically targeted towards safety management. It displays the current safety status of a construction site using the Ashvin safety KPIs published in D4.1.

2 INFORMATION REQUIREMENTS OF THE ASHVIN SYSTEMS

2.1 Ashvin platform

As described above, the Ashvin platform implements the main functionality that allows for collecting IoT related data from construction site and convert this data into meaningful information that can be consumed by different tools. To this end, APIs are required that make this information readily available.

2.2 Simulation-based real-time construction site and logistics planning tool

Figure 2 depicts the required information flow for the DES tool. To allow for the simulation of future construction activities using past productivity data collected by the Ashvin platform, a user in a first step needs to be able to understand the construction activities for the specific site that are part of the site's digital twin stored in the platform. Using these activities different ways to structure the work can be configured by combining the activities within different workflows. Additionally, different scenarios can be set-up for simulating the various configured workflows, accounting for example for different weather conditions. As the platform tracks construction activities together with different environmental conditions, information about different conditions that have been tracked in the past and that are available for simulating future work needs to be provided to the DES tool. For the simulation of the work processes itself stochastic information about the productivity of past processes need to be obtained in the form of probability density functions (pdfs). These pdfs can be conditioned on specific environmental conditions.

After simulation of different construction workflows under various conditions, construction managers can evaluate the different simulated possibilities. This will allow them to reconsider the current activity sequence that is maintained by the platform. Therefore, an API function that allows the DES tool to update the as-planned construction schedule stored within the digital twin needs to be provided.

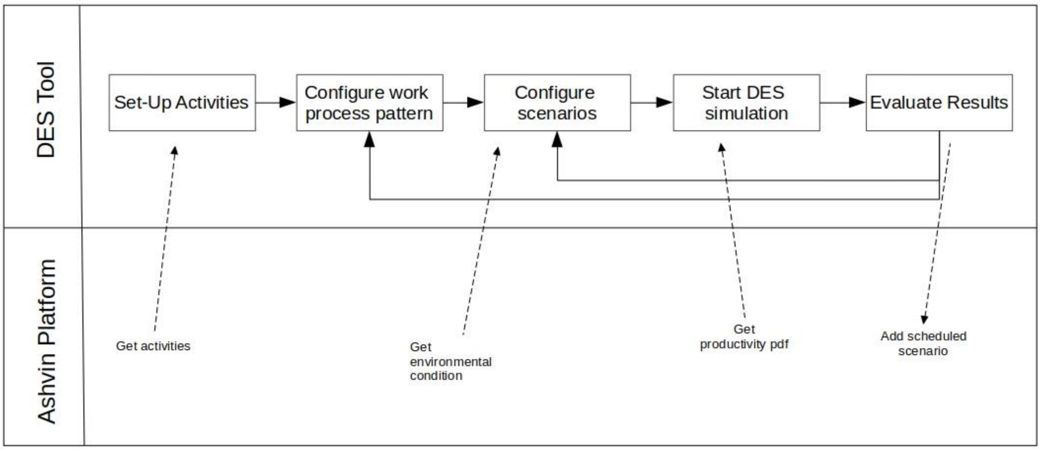

Figure 2: Information flow of the DES tool

2.3 CMT - A configuration management tool to track as-designed and as-built, as well as, to allow for seamless commissioning (developed in T4.5)

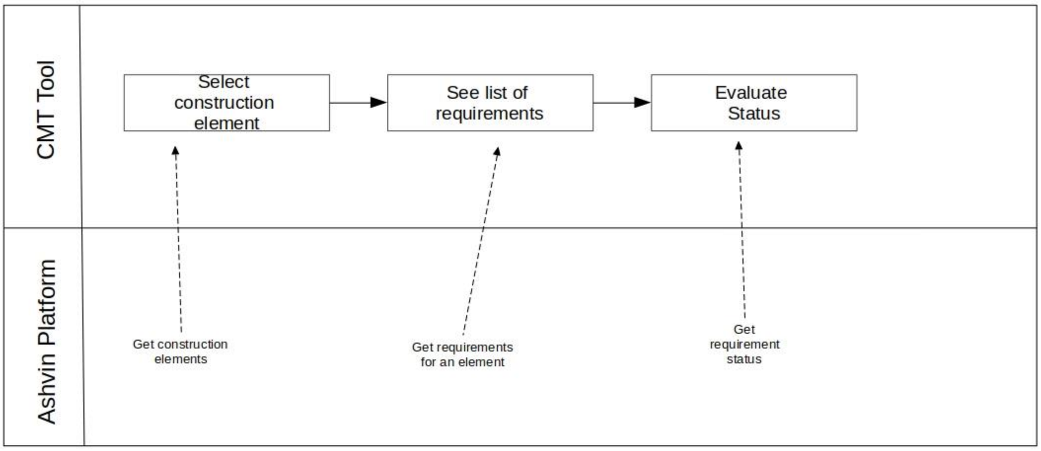

Figure 3: Information flow of the CMT tool

Figure 3 depicts the required information flow for the CMT tool. The purpose of the tool is to allow for dedicated configuration management. To this end, the tool allows for selecting the different elements of a project's building or facility. In a next step, users can select a specific construction element, such as a concrete column, a piece of rebar, a window, or even a power outlet, for understanding the specific requirements of this element. Finally, the tool needs to obtain the actual status of the specific element and how it relates to the initial requirements. As the CMT tool is a visualization and decision-making tool, it does not need to feed information back to the platform.

2.4 Construction monitoring tool with productivity and safety KPI decision making dashboard (4DC-V tool)

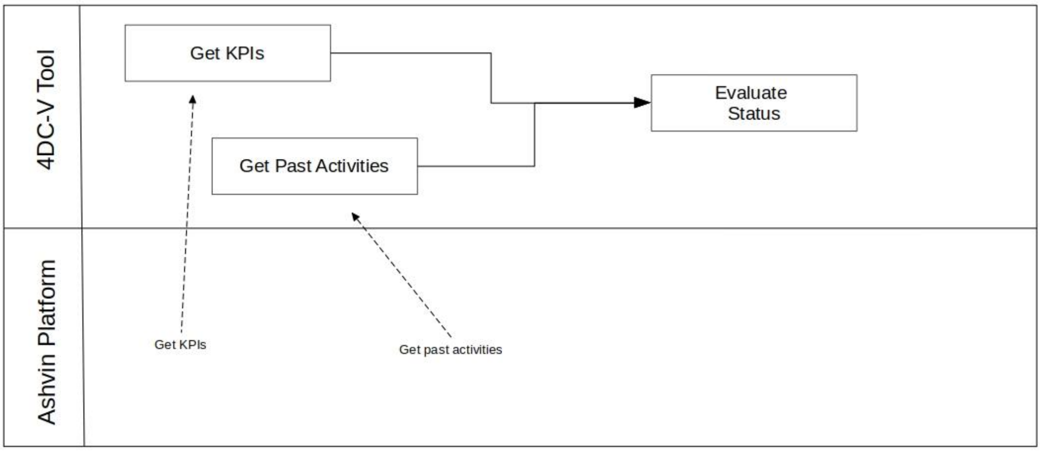

Figure 4: Information flow of the 4DC-V tool

Figure 4 depicts the information tool for the 4DC-V tool for construction monitoring. The tool needs to receive a set of project KPIs and the list of past activities for a project. Together this information then allows a project manager to evaluate the current status of a project based on the KPIs. Moreover, project managers can understand the true past construction progress of the project to understand the KPIs better.

2.5 Privacy ensuring safety management, simulation, and training tool

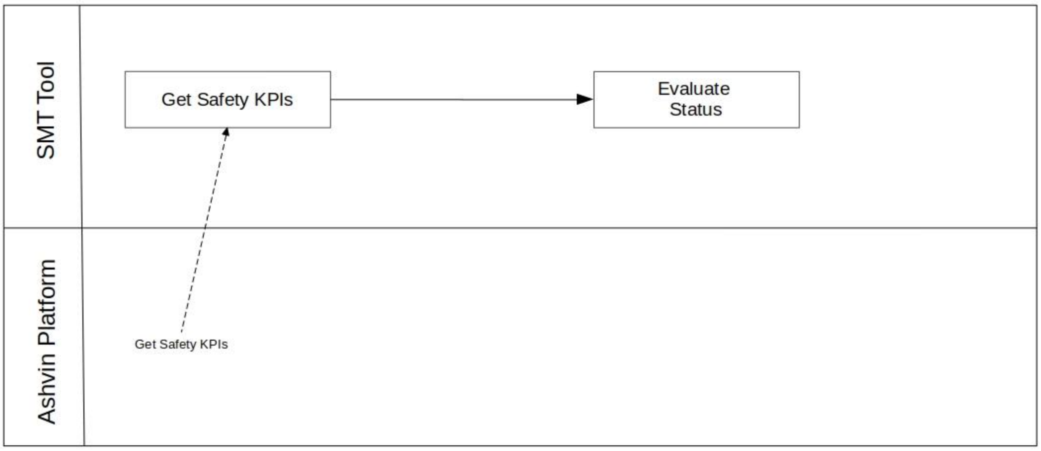

Figure 5: Information flow of the SMT tool

The final safety management tool has the simplest workflow of all the four presented tools (Figure 5). The tool simply needs to obtain a list of the safety related KPIs for suitable representation in a safety dashboard.

3 API FRAMEWORK FOR AUTOMATED EXCHANGE OF INTERPRETABLE DATA BETWEEN SYSTEMS

Based on the above described four Ashvin tools, we devised an API that can provide the information for the above tools. As good API design focuses on providing the minimal amount of data exchange functions, we tried to reduce the API to only the most important specifications. During implementation any API provided by a digital twin platform can of course be extended by more functionality, however, we propose that any digital twin platform for construction provides the minimum set of function specified.

The designed API proposes a simple data model to manage activities, partial differential function for productivity rates, requirements and their status, as well as KPIs. Additionally, the API proposes a set of functions to get the above information for a project.

To make the API widely available, we designed the API using the open standard OpenAPI Specification (OAS) 3.1.0. The specification defines a standard, programming language independent format to specify web-enabled (HTTP) APIs. This allows software developed, independent on their background and language, to understand the capabilities of a service without requiring access to code that implements a specific API.

Interested readers can access the API on the Ashvin open research data repository:

https://zenodo.org/record/7127785

The interactive API documentation is available via the Ashvin website:

https://api-spec.ashvin.eu

The entire API documentation is also appended to this deliverable.

4 RECOMMENDATIONS FOR API DEVELOPMENT

4.1 Implementation of a geometry server

Next to the API itself, we propose that digital twin platforms develop a geometry server that allows them to dynamically serve all geometrical information of a project that would be required by the different representation and visualization needs of Ashvin tools. This geometry layer could for example serve building geometry in one of the available geometry or CAD data exchange standards, such as

- STEP ISO 10303 Standard for the Exchange of Product model data
- The Virtual Reality Modelling Language (VRML)
- Extensible 3D (X3D)

Such a geometry server can be integrated with the Ashvin API simply by providing a reference of API objects with a set of geometry objects served. A simple reference via a unique identifier for geometry objects will allow developers to program 3D supported user interfaces.

Important to realize is, however, that a suitable geometry server for digital twin construction data needs to go beyond simply serving the geometrical description of a building, as it is customary for existing BIM server. A sound construction digital twin geometry server needs to serve a geometrical representation of an entire construction site, including temporary construction objects, construction equipment, construction workers, or flexibly configurable workspaces. More information of required objects can be extracted also from Deliverable 1.2 – An ontology for digital twin models for the construction industry.

4.2 Implementation of the API service layer

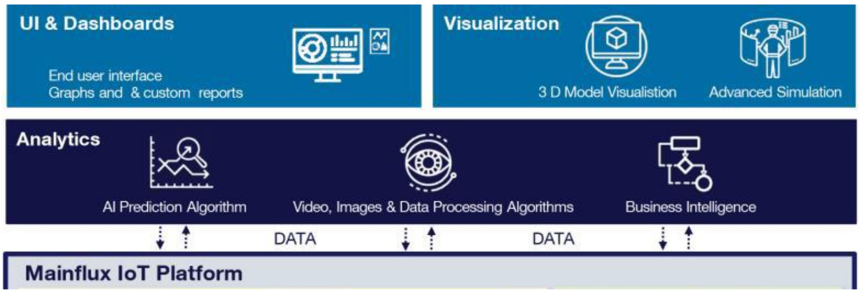

Figure 6: Top layer of the Ashvin digital twin platform for data analytics

The main purpose of the Ashvin digital twin platform is to receive all sorts of data from IoT enabled devices and convert this data to actionable information for informing construction site management. Most of this conversion is conducted within the Analytics layer of the platform (see also Figure 6) that consists of dedicated sets of methods for data fusion, data mining, and data analysis for calculating productivity, KPIs, and construction element conditions. The Ashvin API represented needs to directly connect to this analysis layer.

Considering that

- an endless possibility of sensor and photogrammetric systems can be installed on a construction site from which data can be consumed by the platform,
- countless different possibilities exist for construction activities and construction elements that can somehow be tracked with such sensor systems, and
- infinite possibilities exist to define requirements, productivity rates, environmental conditions and KPIs for construction sites

an endless amount of possible data fusion, mining, and analytics functionality is possible.

Therefore, we recommend designing the architecture of the analytics layer using the object-oriented strategy pattern for managing an ever changing and growing amount of services within the analytics layer. The strategy layer defines a fixed interface for a set of infinite algorithmic strategies. To support the Ashvin API, we would suggest that a minimum of three different strategy patterns are implemented for providing standard interfaces to manage productivity, KPI, and status calculation functionality. This will then allow for the flexible extension and implementation of digital twin platforms for different types and even specific construction sites. At the same time, the patterns will allow for implementing the above proposed API in a standardized manner.

Figure 7 provides an illustrative example of a strategy pattern implementation for different manners to calculate productivity related probability density functions based on three Ashvin demonstration projects (refer to Deliverable 4.2 Discrete Event Simulation Formalism for Productive, Resource Efficient, and Safe Construction Planning for details for the project specifics and the calculations).

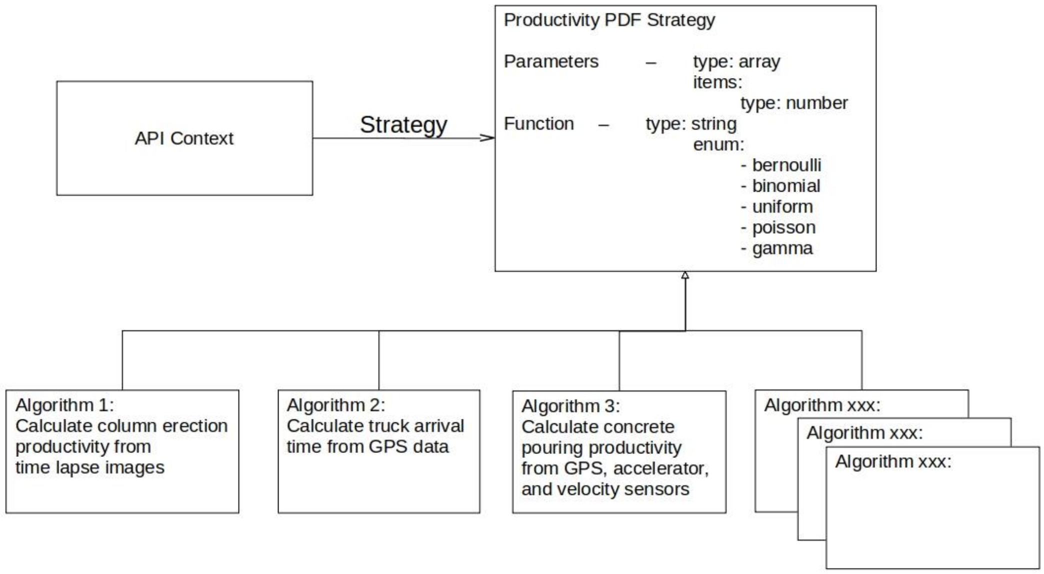

Figure 7: Implementation of a strategy pattern for serving various methods for calculating probability density functions representing productivity rates

5 EXISTING STANDARDS FOR EXCHANGING MAINTENANCE INFORMATION

As previously mentioned, this report focuses on describing the background for the Ashvin API that has been developed with the focus of supporting the construction phase. However, maintenance of the built assets comes logically as a next phase to support various data-driven decisions based on various monitoring activities. Yet, maintenance phase was one of the first areas of focus of the research community and significant advances were made towards standardizing the exchange of asset's monitoring information. Moreover, quite significant details were provided in deliverable 5.1. Hence, we only present here the existing standards within the field.

Theiler et al (2020) provides a state-of-the-art overview of

- SHM-related regulations, standards, and guidelines (presented in Table 1)
- Metamodels architectures (ontologies and description languages)

meant to support the realization of the digital twins for maintenance.

Table 1: - SHM-related regulations, standards, and guidelines

Regulations	Guidelines	Standards
RVS 13.03.01:2012: Monitoring of bridges and other civil engineering structure	RUV:2008: Guideline for monitoring of road safety of federal structures FHWA-HRT-16-007: Long-term bridge performance (LTBP) program protocols, version 1 FHWA-HRT-09–040: State of the practice and art for structural health monitoring of bridge substructures F08b: Guideline for structural health monitoring Development of a model health monitoring guide for major bridges DWA-M 514:2011: Dam Surveillance	DIN 1076:1999-11: Engineering structures in connection with roads – Inspection and test GB 50982-2014: Technical code for monitoring of building and bridge structures BD 63/17: Highway structures: Inspection and maintenance Part 4

The SHM-related regulations, standards, and guidelines presented in Table 1 streamline various processes and aspects related to: (1) data acquisition, (2) data communication, (3) data processing, (4) data storage, (5) data interpretation, (6) data retrieval, and some other generally applicable. For example, MatchFEM tool functionality from Ashvin toolkit falls under the category data interpretation.

As metamodels, the authors identify UML 2 and EXPRESS as most prominent. On an earlier paper (Fitz et al, 2019), the technical specifications of data exchange for SHM are specified using these.

6 CONCLUSION

This report presents, in the form of API, the extract, transform, and load (ETL) processes that enable information exchange for the development of digital twins for the construction sites that form the cornerstone of the ASHVIN platform. Moreover, the report provides a set of IDM (Information Delivery Manuals) for the identified ETL processes of the various Ashvin systems. While at first these were developed around Ashvin systems, the final interoperability concepts proposed in this report are meant to generalize and to provide a solid foundation to various actors and serve as a cornerstone for future standard development endeavours.

7 REFERENCES

Fitz, T., Theiler, M. and Smarsly, K., 2019. A metamodel for cyber-physical systems. Advanced engineering informatics, 41, p.100930.

Theiler, M., Ibáñez, S., Legatiuk, D. and Smarsly, K., 2020. Metaization concepts for monitoring-related information. Advanced engineering informatics, 46, p.101158.

8 APPENDIX A – API DEFINITION

File ▾ Edit ▾ Insert ▾ Generate Server ▾ Generate Client ▾

Ashvin Construction Digital Twin Interoperability API 0.0.1 OAS3

default

GET /activities

Returns a list of activities

Parameters

Try it out

No parameters

Responses

Code	Description	Links

Code	Description	Links
200	Everything went fine - an array of all activity IDs Media type application/json Controls `Accept` header. **Example Value** Schema `[ "string" ]`	*No links*

POST **/activities**

Let's a user post a new description

Parameters

Try it out

No parameters

Request body

application/json

Example Value Schema

```
{
  "id": "string",
```

```
  "name": "string"
}
```

Responses

Code	Description	Links
200	Successfully added a new Activity Media type application/json Controls `Accept` header. **Example Value** Schema `{ "id": "string", "name": "string" }`	*No links*

PUT /linkactivity/{activityID}/{MFXThingsID}

Links an activity to a Mainflux THING

Parameters

Try it out

Name	Description
activityID * required `string` *(path)*	activityID
MFXThingsID * required `string` *(path)*	MFXThingsID

Responses

Code	Description	Links
200	Successfuly associated the activity with the Mainflux THING	*No links*

GET `/things/{activityID}`

Get the things connected to an activity

Parameters

Try it out

Name	Description

Name	Description
activityID * required `string` *(path)*	activityID

Responses

Code	Description	Links
200	Returns a list of things IDs	*No links*

GET `/getpdf/{activityID}`

Get a probability density function for an activity

Parameters

Try it out

Name	Description

Name	Description
activityID * required `string` *(path)*	activityID

Responses

Code	Description	Links
200	returns the pdf Media type **application/json** Controls `Accept` header. **Example Value** Schema	*No links*


```
{
  "function": "Bernoulli",
  "parameters": [
    0
  ]
}
```


GET **/calculatepdf/{activityID}**

recalculates the pdf of an activity and returns the new pdf

Parameters

Try it out

Name	Description
activityID * required string *(path)*	activityID

Responses

Code	Description	Links
200	returns the recalculated pdf Media type **application/json** Controls `Accept` header. **Example Value** Schema	*No links*

```
{
  "function": "Bernoulli",
  "parameters": [
    0
  ]
}
```

PUT **/addsimulatedtask/{scenarioID}**

adds a simulated activity as scheduled task to a schedule scenario

Parameters

Try it out

Name	Description
scenarioID * required string *(path)*	scenarioID

Request body

application/json

Example Value Schema


```
{
  "id": "string",
  "name": "string",
  "start": "2022-09-16T16:25:05.617Z",
  "end": "2022-09-16T16:25:05.617Z"
}
```


Responses

Code	Description	Links
200	successfully added a scheduled task to the scenario	*No links*

POST **/schedulescenarios**

Set-Up a new schedule scenario

Parameters

Try it out

No parameters

Request body

application/json

Example Value Schema


```
{
  "id": "string",
  "name": "string",
  "activities": [
    {
      "id": "string",
      "name": "string",
      "start": "2022-09-16T16:25:05.620Z",
      "end": "2022-09-16T16:25:05.620Z"
    }
  ]
}
```

Responses

Code	Description	Links
200	Successfully added a new ScheduleScenario	*No links*

Media type

application/json

Controls `Accept` header.

Example Value Schema


```
{
  "id": "string",
  "name": "string",
  "activities": [
    {
      "id": "string",
      "name": "string",
      "start": "2022-09-16T16:25:05.622Z",
      "end": "2022-09-16T16:25:05.622Z"
    }
  ]
}
```


GET **/schedulescenarios**

Returns the scheduled scenarios

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	the list of all scheduled scenarios	*No links*

Media type

application/json

Controls `Accept` header.

Example Value Schema


```
[
  {
    "id": "string",
    "name": "string",
    "activities": [
      {
        "id": "string",
        "name": "string",
        "start": "2022-09-16T16:25:05.624Z",
        "end": "2022-09-16T16:25:05.624Z"
      }
    ]
  }
]
```

GET /constructionElement

Returns a list of construcion elements

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Everything went fine - an array of all construction elementIDs Media type application/json Controls `Accept` header. **Example Value** Schema	*No links*

```
[
  "string"
]
```

POST /constructionElement

Let's a user post a new construction element

Parameters

Try it out

No parameters

Request body

application/json

Example Value Schema


```
{
  "id": "string",
  "name": "string"
}
```


Responses

Code	Description	Links

Code	Description	Links
200	Successfully added a new construction element Media type application/json Controls `Accept` header. **Example Value** Schema `{ "id": "string", "name": "string" }`	*No links*

PUT **/linkactivity/{constructionelementID}/{MFXThingsID}**

Links a construction element to a Mainflux THING observing the status of the component

Parameters

Try it out

Name	Description
constructionelementID * required `string` *(path)*	constructionelementID

Name	Description
MFXThingsID * required string *(path)*	MFXThingsID

Responses

Code	Description	Links
200	Successfuly associated the construction element with the Mainflux THING	*No links*

GET `/things/{constructionelementID}`

Get the things connected to a construction element

Parameters

Try it out

Name	Description
constructionelementID * required string *(path)*	constructionelementID

Responses

Code	Description	Links
200	Returns a list of things IDs	*No links*

GET `/getcomponentstatus/{constructionelementID}`

Gets the automatically calculated status of a construction element

Parameters

Try it out

Name	Description
constructionelementID * required `string` *(path)*	constructionelementID

Responses

Code	Description	Links
200	returns the pdf	*No links*

returns the pdf

Media type

application/json

Controls `Accept` header.

Example Value Schema

```
{
  "requirement": "string",
  "status": "string",
  "quantifiedstatus": 0
}
```

GET **/calculatestatus/{constructionelementID}**

recalculates the status of a construction element and returns the new status

Parameters

Try it out

Name	Description
constructionelementID * required string *(path)*	constructionelementID

Responses

Code	Description	Links
200	returns the recalculated status	*No links*

Media type

application/json

Controls `Accept` header.

Example Value Schema

```
{
  "requirement": "string",
  "status": "string",
  "quantifiedstatus": 0
}
```

Schemas

```
Activity {
   id                    string
   name                  string
}

Pdf {
   function              string
                         Enum:
                           [ Bernoulli, Binomial, Uniform, Poisson, Gamma ]
   parameters              [number]
}

ScheduleScenario {
   id                    string
   name                  string
   activities              [ScheduleTask {
                           id                    string
                           name                  string
                           start                 string($date-time)
                           end                   string($date-time)
                         }]
}
```

```
ScheduleTask {
    id              string
    name            string
    start           string($date-time)
    end             string($date-time)

}
```

```
ConstructionElement {
    id              string
    name            string

}
```

```
ElementStatus {
    requirement       string
    status            string
    quantifiedstatus  number

}
```