DAFNI as a Digital Twin Platform

DAFNI Champion: Cristian Genes

Lead Contact: Daniel Coca

Affiliation: Department of Automatic Control & Systems Engineering, University of Sheffield

1. Introduction

1.1 Background

Recent advances in internet of things technologies, data analytics, artificial intelligence, and cloud computing enabled new possibilities for modelling in general, and infrastructure modelling in particular. As the infrastructure systems are becoming increasingly cyber-physical and interconnected, there is a pressing need to develop next-generation systems engineering tools and methods for the design, development and operation of the future intelligent systems-of-systems.

A radical new approach is to use the vast amounts of data collected from manufacturing, maintenance, operations, and operating environments to create 'Digital Twins' i.e. models of specific assets, processes or systems which can be used to mirror in real-time the operation of their physical counterpart.

A Digital Twin is more than just a mathematical model of a physical system or process. It changes dynamically in near real-time as the state of the physical object changes by using environmental and operational data that is consistently acquired and assimilated, thus enabling physical infrastructure assets to 'live a parallel life' in the virtual world for their entire life-cycle. The Digital Twin is used in combination with analytics workflows to assess the consequences of various design choices, to determine optimal actions that maximise key performance metrics, evaluate alternative development and implementation strategies and to forecast the effects of operation and servicing decisions.

Data & Analytics Facility for National Infrastructure (DAFNI) is a unique, key technology platform currently under development, which will enable the development of Digital Twins of infrastructure systems, assets and processes. This £8 million investment will provide massive secure data storage, awesome computer power and the next generation of tools and services for model simulation, data capture and assimilation, machine learning and visual analytics that will be continuously refined and updated.

1.2 Aims and Objectives

The aim of the project is to demonstrate the unique, advanced technological capabilities that make DAFNI an ideal platform for collaborative development, validation and implementation of Digital Twins for large-scale complex systems. To that end, the project implements a pilot study that helps evaluate the existing capabilities of the platform and informs on new functionalities required for accommodating full-scale digital twins.

1.3 Summary of Achievements

The main achievement of the project is the pilot study that implemented a real-time traffic forecasting model for over 640 sensors in Sheffield. The pilot is the foundation for a future Sheffield traffic digital twin and strengthens the links between DAFNI and the Sheffield Urban Observatory by enabling models on DAFNI to use real-time data from the observatory.

In addition, the pilot study enabled the evaluation of DAFNI's readiness level and highlighted new functionalities that are required for accommodating digital twins on the platform.

An online workshop focused on Digital Twins was organised with the DAFNI technical team and researchers at the University of Sheffield. The event presented some of the challenges and opportunities in digital twin research, the results of the pilot study, and introduced the DAFNI platform to researchers in Sheffield and to the wider digital twins community.

2. DAFNI Evaluation Framework

2.1 Digital Twins definitions

There are several digital twin definitions available in the literature but there is no universally accepted formulation. Different domains have different requirements for a model to become a digital twin and some definitions are more specific (in terms of functionalities required) than others. For example, IBM defines a digital twin as "a virtual representation of a physical object or system across its lifecycle, using real-time data to enable understanding, learning and reasoning"[1]. In the IBM definition, the need for real-time data is clearly specified. In addition, the digital twin needs to enable a better understanding of the modelled system and help in the decision making process. This process is not only temporary but it needs to happen across the lifecycle of the system.

Another example of digital twin definition is from Siemens: "A digital twin is a virtual representation of a physical product or process, used to understand and predict the physical counterpart's performance characteristics"[2]. Clearly there is a significant overlap between the two definitions but the Siemens one does not specify requirements such real-time data or the need to use the twin across the lifecycle.

There is also a range of definitions coming from academia. The Cambridge Centre for Digital Build Britain (CDBB) defines digital twins as “realistic digital epresentations of physical things. They unlock value by enabling improved insights that support better decisions, leading to better outcomes in the physical world”[3]. The focus in the CDBB definition is on the outcomes and the opportunities enabled by digital twins rather than requirements.

Furthermore, Michael Batty proposes the following definition: “A digital twin is a mirror image of a physical process that is articulated alongside the process in question, usually matching exactly the operation of the physical process which takes place in real-time”[4]. Similar to the IBM definition, the real-time aspect included in this definition as well as the idea that the twin follows the system across its lifecycle. Although there is no complete definition that is accepted by all entities involved, it is clear that there is a big overlap in terms of requirements for a model to be a digital twin. Specifically, the model needs to be an accurate, real-time or right-time representation of the modelled physical object of phenomena that enables better understanding of the process modelled and can be used for better decision making. In view of this, the resulting Digital Twin ecosystem is depicted in Figure 1.

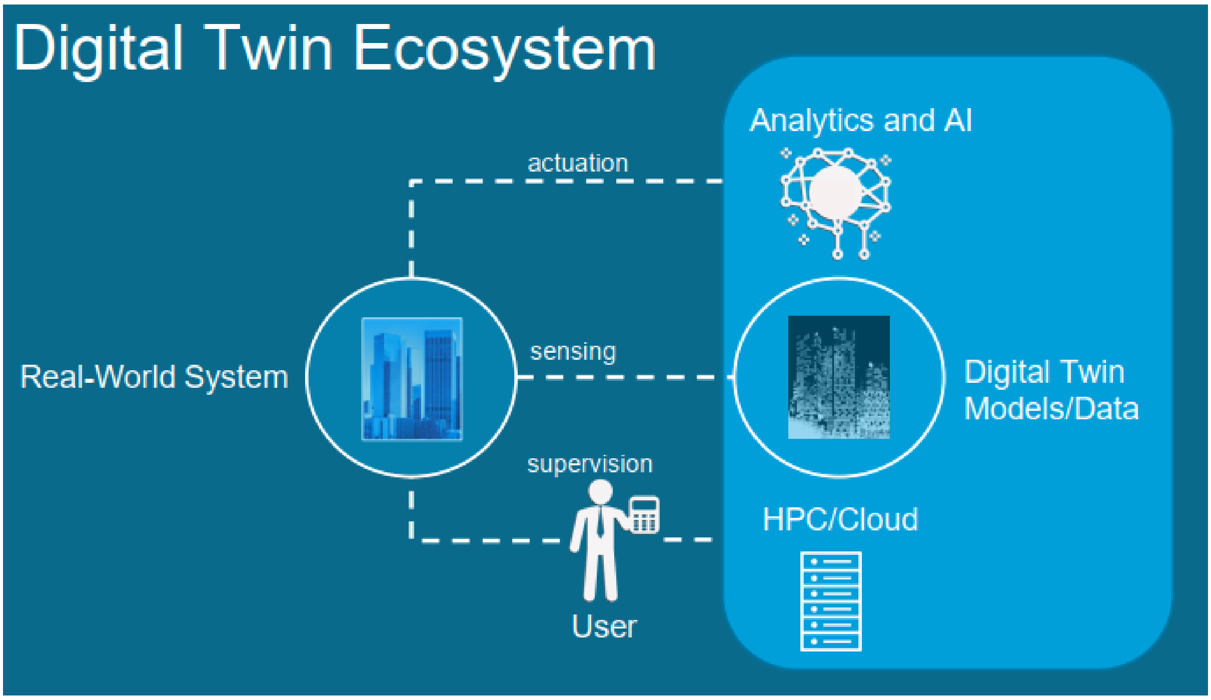

Figure 1. Digital Twin Ecosystem.

The requirements for a model to become a Digital twin, also translate into a set of software and hardware requirements for any digital twin enabling platform. The following section discusses the platform requirements in more detail.

2.2 Requirements for Digital Twin implementation

The process of implementing a full-scale Digital Twin poses significant software and hardware challenges for the platform hosting the twin. First, the security aspects need to be considered and access to data and models used needs to be managed and controlled. Digital twins incorporate a number of models and datasets that work different scales and some of these datasets and models might be proprietary. In certain cases an accurate representation of the modelled object/phenomena might require access to proprietary data and models. For that reason, the security of the platform is a key requirement for encouraging researchers and private companies to share their data and models.

A full-scale digital twin is likely to incorporate a federation of twins that work in different domains and at different scales[3]. Therefore, creating a digital twin is a multidisciplinary problem which requires collaboration across multiple disciplines. A key enabler of multidisciplinary work is a platform that allows users to share datasets and models, and combine them into complex workflows. Furthermore, different research areas use different software packages to create models. For that reason, the collaborative platform needs to be language agnostic, i.e., it should enable the integration of models written in different programming languages. The platform should also store or facilitate easy access to datasets from various research domains.

Digital twins require significant computational power. Because of their federated nature[3], efficient parallel processing is paramount for accommodating full-scale digital twins. In addition, real-time processing also requires fast and reliable access to data.

In operation, digital twins interact with users and display results through visualisation tools. Therefore, the implementation of DTs requires flexible and versatile tools for creating visualisations. Furthermore, the user interface should enable bidirectional interaction with the model. For example, different modelling scenarios can be assessed from the visualisation tool which enables better decision making.

Another requirement that is mentioned is some DT definitions real-time data. While in certain cases it might not be possible or necessary, most models need to be updated in real or near real-time to accurately track the performance of the system they model. The platform hosting the twin should store or provide access to real-time data streams which can easily be integrated in models and workflows.

To sum up, the Digital twin enabling requirements are:

- Secure storage of data and models
- Enable access to real-time and historical data
- Collaborative environment for sharing data and models within research groups
- Support for a wide range of programming languages
- Seamless integration of multiple datasets and models into complex workflows
- High Performance Computing capabilities
- Optimized parallel processing
- Support for visualization tools
- Interactive user interface

Most of these functionalities are already available on DAFNI and the rest are under development. The following section describes the existing functionalities in more detail.

2.3 DAFNI's current level of readiness

DAFNI is a collaborative platform for high performance computing that supports infrastructure research and long term planning. The platform allows users to store both datasets and models and combine them into complex workflows. In addition, it offers great flexibility in terms of types of data stored and the programming languages it supports. DAFNI can run and integrate models written in any programming language. The trade-off is a slightly more complex uploading process in which the user needs to containerise the model. To help with this process, a number of examples and tutorials are available on DAFNI's website.

The access to the platform is controlled and users can create their own groups with which they can share datasets and models. This makes DAFNI a great collaborative platform for multidisciplinary research.

The collaborative aspect as well as the data and models flexibility makes DAFNI a key enabler of digital twin research. The following functionalities are already available on DAFNI and are essential for digital twin research and implementation:

- Secure storage of data and models
- Access to historical data
- Collaborative environment for sharing data and models within research groups
- Support for a wide range of programming languages
- Seamless integration of multiple datasets and models into complex workflows
- High performance computing - to support higher level of detail and accuracy for large infrastructure systems
- Support for visualization tools - allow users to create custom visualisation tools for their models

However, these functionalities are not sufficient to accommodate full-scale digital twins. The following section details some of the new functionalities required on DAFNI to fully accommodate digital twins.

2.4 New functionalities required

An accurate representation of the modelled system is a key element of every digital twin definition. This requires access to timely and accurate data from the system. For that reason, access to real-time data is essential for digital twin enabling platforms. DAFNI is currently working on integrating real-time data streams into the models. This is enabled through collaborations with Urban Observatories.

A full-scale digital twin is likely to incorporate a federation of twins that work in different domains and at different scales[3]. Therefore, optimized parallel processing is another functionality required for any digital twin enabling platform. In addition, to help with decision making, digital twins need to compute a large number of simulations with different input parameters. DAFNI is currently working on expanding the parallel processing features on the platform. For example, an iterator element is available to test the sensitivity of the model to different input parameters. However, the user can only select the interval in which the parameters take values and not the actual values that are of interest.

Another essential element of a digital twin is the user interface. Digital twins need to be user friendly and allow complex analysis, decision making and optimization without accessing and modifying the source code. In view of this, the visualisation of a digital twin needs to allow real-time updates and facilitate the simulation of different scenarios. Existing visualisations on DAFNI can only be used to display the results produced after running the model. The results displayed cannot be updated and new simulation scenarios cannot be tested from the visualisation. New functionalities to enable interactive visualisations are needed to enable the implementation of digital twins on DAFNI.

3. DAFNI Digital Twin Pilot

3.1 Objective of the pilot study

The pilot study is designed to test the existing capabilities of the DAFNI platform for digital twins and identify new functionalities that are required for accommodating full-scale digital twins. In collaboration with the Sheffield Urban Observatory, the pilot study uses real-time traffic data from over 640 sensors and predicts the evolution of traffic and where congestion might occur. The software implementation is done in MATLAB and therefore, the first step is to enable MATLAB-based models to run on DAFNI. The following section presented the steps required to upload and run a MATLAB model on DAFNI.

3.2 Matlab models on DAFNI

DAFNI is a programming language agnostic platform, i.e. it can run models written in any programming language provided that they are integrated in a self contained Docker container. MATLAB is a widely used programming language in engineering research. One of the first challenges of this project was to enable MATLAB-based models to run on DAFNI. Because MATLAB is a licence-based product, building a container with the MATLAB script and the MATLAB software will require each user that wants to run the code to have their own MATLAB licence. Furthermore, the way the Docker containers are stored on DAFNI makes the process of adding a MATLAB licence into an uploaded container very difficult.

To overcome the MATLAB licence limitation, the uploaded container includes MATLAB Runtime and the executable version of the MATLAB code. Because DAFNI is a Linux-based platform, the executable version of the MATLAB code needs to be created in Linux. To sum up, creating a DAFNI ready MATLAB-based container requires the following two steps:

- Create the Linux executable version of the MATLAB code
- Build a Docker container with MATLAB Runtime and the Linux executable file from the first step

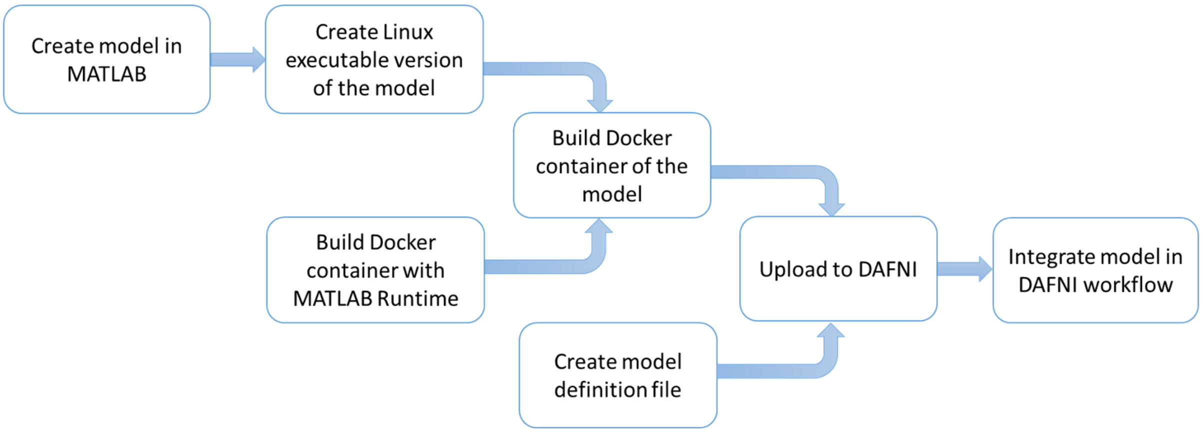
Create model in MATLAB
Create Linux executable version of the model
Build Docker container of the model
Build Docker container with MATLAB Runtime
Upload to DAFNI
Integrate model in DAFNI workflow
Create model definition file

Figure 2. Uploading process for MATLAB models on DAFNI.

The rest of the uploading process is depicted in Figure 2 and described in more detail on DAFNI's website[5], i.e. create the model definition file using the instructions provided and upload the container and the model definition file onto DAFNI. The following section describes the model built for the pilot study in more detail.

3.3 Sheffield traffic digital twin model

This section presents the traffic forecasting model that is the foundation for a future DAFNI-powered Sheffield Traffic Digital Twin. The MATLAB model uses historic and real-time traffic data from more than 640 sensors in the Sheffield area and predicts the evolution of the traffic and where congestion might occur. The results are displayed in a bespoke visualisation created in Jupyter Notebook.

3.3.1 Sheffield traffic data

The Sheffield Urban Observatory harvests traffic flow (number of cars per minute) data since July 2019. The traffic prediction model uses historical traffic data since September 2019. There are more than 640 sensors deployed in the Sheffield area and each sensor reports the traffic flow every 5 minutes. The data is available through an API call from the Sheffield Urban Observatory servers.

To exploit the periodicity of traffic data, the model stores data in vectors that cover one sensor and one day of traffic (24 hours interval). This enables quick comparisons between different days and different locations in the city. In this setting, the historical data is organized in a matrix format in which different rows cover different time intervals during the day and each column corresponds to a particular sensor and a particular day. Hence, the matrix has 288 rows and the number of columns is equal to the number of days times the number of sensors. In practice, the number of columns is smaller because some of the sensors are temporarily inactive (due to faults or roadworks) or they report traffic values that are not consistent with the other sensors (outliers). A data quality check and pre-processing step is included to guarantee the quality and quantity of data before it is used to calculate new predictions.

3.3.2 Data quality and pre-processing

The traffic forecasting model uses two types of traffic flow data. First is the historical data between September 2019 and the day before the prediction, and second is the real-time data that contains the latest traffic values collected from the sensor of interest. The historical and real-time traffic data go through the same quality checks and pre-processing steps to ensure consistency. For the historical data, each column of the data matrix is individually assessed and if the data quality checks are not passed, the column is removed from the matrix. There are four main steps in this process:

- Identify active sensors - find sensors that are inactive and do not report any traffic flow values.
- Outlier detection - identify sensors that report values that are not consistent with the other sensors, example: sensors that report constant values over long periods of time
- Missing data - identify sensors that have more than 10% missing data over a 24 hour period.
- Smoothing - apply a centred moving average of length 4 (20 minutes time window) on all columns of the data matrix that passed the quality checks

After all the pre-processing steps, the data matrix will have a smaller number of columns because it only contains reliable and accurate data from the traffic sensors. The historical data matrix needs to be updated regularly to contain pre-processed data up to the day before the prediction. Therefore, the same sequence of checks and pre-processing steps are applied to all the new columns that are attached to the data matrix in the updating process. A separate MATLAB model is built to this but it is not uploaded on DAFNI yet due to some functionalities required not being available. For example, the model cannot be set to run automatically every 24 hours and once the model updates the historical dataset, the new dataset cannot be automatically set as an input to the traffic prediction model.

In addition to the historical data, the forecasting algorithm also uses real-time data, i.e. latest values reported by the sensors for which the prediction is calculated. The data is accessed through the same API call and goes through the same quality checks and pre-processing steps. The main difference here is that if the data quality checks are not passed the prediction cannot be calculated. In this case the model stops and returns an error message. The following section explains in more detail how the prediction is being calculated.

3.3.3 Traffic forecasting model

The traffic forecasting model uses both the historical data and real-time data after they pass the quality checks and pre-processing steps to calculate the predicted traffic flow for a particular sensor. The model has two input parameters: the sensor id and the length of the prediction horizon (in minutes). To facilitate the explanation of the prediction process, let us assume that the current time is 2 PM and the model is used to estimate the traffic between 2 PM and 3 PM. A vector with the latest traffic data for the sensor of interest is downloaded from the API call and goes through the pre-processing steps presented in Section 3.3.2. The length of this vector, i.e. the number of recent values used for prediction is currently set to 96 which corresponds to the last 8 hours of data. Therefore, in our scenario, the vector with real-time data contains the values between 6 AM and 2 PM.

The next step in calculating the prediction is to truncate the matrix with the historical pre-processed data to cover the same time window, i.e. 6 AM to 2PM. This allows a direct comparison between the traffic pattern observed today and the previous traffic patterns in the historical data, i.e., from different parts of the city and different days in the past. A selection algorithm is used to select a subset of the columns from the truncated historical pre-processed data matrix for which the traffic patterns are correlated with latest measurements. Only this subset of columns from the historical data is used in the following steps. Specifically, a machine learning regression model is trained to estimate the correlation between historical data and the most recent traffic values. The weights of the regression model are adjusted to predict today's traffic data using the selected historical data from the same time period, i.e. 6 AM to 2 PM. Furthermore, for the selected historical data, traffic values between 2 PM and 3 PM (the prediction window) are also available. Based on the assumption that if the traffic correlated between 6 AM and 2 PM ts will also correlate between 2 PM to 3 PM interval, the predicted values are calculated using the trained machine learning regression model using the selected historical data between 2 PM and 3 PM as input. Different types of regression models have been assessed and the best performance was obtained for the Gaussian Process Regression. In addition, the computation time is around one second for one sensor which enables us to scale this up using DAFNI's HPC resources. However, new DAFNI functionalities are required to be able to run parallel sessions of the same model with different input parameters. More details about this are presented in Section 3.3.5.

The following section presents the bespoke visualisation created for the traffic forecasting model.

3.3.4 Visualisation

A key element of any Digital Twin is the visualisation. For this project, a bespoke visualisation was created in Jupyter Notebook. A new, more intuitive way of building visualisations is currently being developed for the DAFNI platform.

Figure 3 shows the visualisation for the case in which the prediction is calculated for multiple sensors and for a prediction horizon of 30 minutes. The interface shows the map of the Sheffield area and at the location of each sensor a coloured circle with the color corresponding to the predicted value of the traffic flow. A slider at the bottom allows the user to change the time at which the predicted value is displayed - from 5 minutes ahead to the value specified as the prediction horizon. That being said, the model is not yet a full-scale digital twin and further work is required to implement additional features. The following section presents a few functionalities that will be added subject to time and new DAFNI functionalities for digital twins.

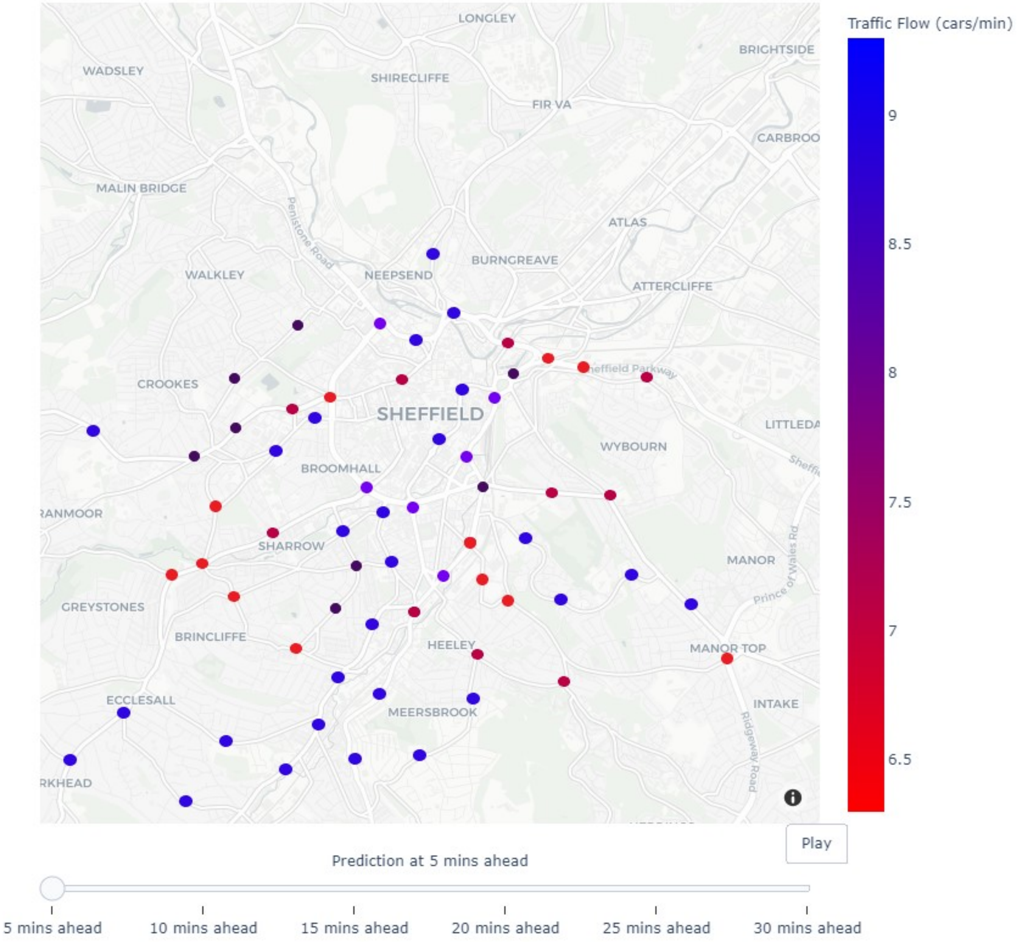

Figure 3. Pilot study visualisation.

3.3.5 Future work

The traffic forecasting model presented in Section 3 is only the foundation of a full-scale digital twin of the Sheffield traffic. More functionalities need to be added to obtain a twin with a higher maturity level.

First, the process of updating the historical pre-processed dataset needs to be automated. This requires a separate model that runs periodically (every day in this case) on DAFNI and updates the historical dataset. This step requires two new DAFNI functionalities:

- Automatic scheduling of model running time (to run the model every day at a specific time)

- Automatically feed the latest version of a dataset into the traffic prediction model every time it is used

In addition, the current version of the model only calculates the predictions and displays the results for one sensor. This process needs to be scaled-up by running multiple parallel sessions of the same code with different input parameters (different sensor ids). This functionality is not yet available on the platform but it will likely be required for other digital twin models.

The main objective of the model is to predict congested areas. Currently, only traffic flow data is used in the model and this type of data is not sufficient for identifying congestion. A congested area has a low traffic flow value but the same applies for cases when the traffic is light (for example during the night time). To distinguish between the two cases, an estimate of the average speed is necessary, such that if the traffic flow is low and average speed is low - there is congestion, and if the traffic flow is low and average speed is high - there few cars on the road. The current version of the model is limited by the traffic data available through the Sheffield Urban Observatory but in the future more types of traffic data such as occupancy will be made available. This will allow the model to be more accurate in identifying congestion.

Finally, one of the key aspects of a full-scale digital twin is the accessibility of the user interface. An interactive visualisation is envisioned to be implemented to allow users to select areas of interest on the map or produce custom output datasets. One of the major drawbacks of the current visualisation is that the user cannot update the predictions. To do that, the model needs to be run again from the DAFNI platform. An interactive interface will allow the user to run the model and plot data of interest for decision makers without manually setting input parameters for the model.

4. Engagement, Advocacy and Dissemination Activities

This project strengthened the links between DAFNI and the Sheffield Urban Observatory by incorporating real-time and historical data into DAFNI workflows. Traffic data from more the 640 sensors is transferred in real-time from the Urban Observatory to the traffic forecasting model on DAFNI. Furthermore, historical pre-processed data and the predictions are available as DAFNI datasets and can be used by other models on the platform.

In addition, a “DAFNI as a Digital Twin Platform” workshop was organised on 26th of February 2021. The event included a number of presentations from the DAFNI technical team, researchers from the University of Sheffield and the Programme Manager for the National Digital Twin Programme. The event addressed some of the main challenges in digital twin research, presented the latest results from the pilot study in Section 3, and introduced DAFNI to researchers from University of Sheffield as well as the digital twins community.

The champion also engaged with the Center for Digital Built Britain (CDBB) and shared DAFNI’s capabilities and vision with other researchers at the “Integrated modelling for built and natural environments" workshop that took place in January 2021. The main outcome of the workshop is a CDBB report that highlights the importance of platforms like DAFNI for enabling digital twin research and collaboration.

Other engagement activities include a webpage and a blog post on DAFNI’s website which enables collaborations with members of the digital twins community through different social media platforms.

References

[1] Armstrong, M. 2020. Cheat sheet: What is Digital Twin? Available here
[2] Siemens website - Digital Twins. Available here
[3] Bolton, A., Butler, L., Dabson, I., Enzer, M., Evans, M., Fenemore, T., Harradence, F., Keaney, E., Kemp, A., Luck, A. and Pawsey, N., 2018. Gemini Principles.
[4] Batty, M. 2018. Digital twins. Environment and Planning B: Urban Analytics and City Science. 45. 817-820. 10.1177/2399808318796416.
[5] DAFNI website - How to Create a DAFNI Ready Model. Available here