Building Experimental Laboratory for Digital Twin in Service Oriented Architecture

Hoa Thi Nguyen
Faculty of Computer Sciences
Østfold University College
Halden, Norway
Department of Informatics
University of Oslo
Oslo, Norway
hoa.nguyen@hiof.no

Øystein Haugen
Faculty of Computer Sciences
Østfold University College
Halden, Norway
oystein.haugen@hiof.no

***Abstract*—The concept of Digital Twins is still in an infant state of development. Digital Twins are often built as a tool to aid in better understanding of physical systems through simulation. They can be used to visualize information during operations and provide instructions during training or execution of procedures. The use of Digital Twins to test is appealing as it can be done quickly and safely. However, testing without inclusion of the physical system can lead to a reality gap. The reality gap can lead to high risks when applying concepts tested on digital Twins to the physical system directly. Sometimes interaction with the physical system is unfeasible. In this paper, we present an experimental laboratory that we built to provide a platform for the development of high quality Digital Twins through a feedback loop. The physical system is a Palfinger crane. Our replicate physical twin is a Universal collaborative industrial robot model UR16e due to its similar anatomy to the crane. The RoboDK simulation software was used to rapidly develop a digital twin of the UR16e. We demonstrate a solution to the interoperability problem in digital Twins using the monitoring adaptation loop from the Autonomic Adaptation System of the Arrowhead Framework.**

I. Introduction

The rapid development of information technologies is driving a revolution in industry, as innovative technologies like artificial intelligence, robotics enable not only automating of manufacturing process, but also automating of knowledge and wisdom. This new wave is put under an umbrella term called Industry 4.0. In this context, Digital Twins [1] (DT) has emerged as a "fusion platform" for multi-disciplinary technology, data science from information space to the manufacturing floor of physical space. The first challenge is connectivity between to different spaces. These entities do not speak the same language leading to the problem of interoperability. Interoperability is in space. Heterogeneous devices use different communication protocols. Stakeholders have different mindsets in different fields. Interoperability is in time. Systems need to evolve due to technological progress. It is required to integrate with legacy systems, while adapting to future upgrading. DT should be flexible, scalable in the sense that it allows easily adding or removing components seamlessly. In this paper we show that Arrowhead Framework can accommodate all the needs for interoperability in DT. We demonstrate with Norwegian use case, a project to build DT for a Palfinger crane mounted on a marine ship. The project involves stakeholders with different interests but can create a synergy DT.

The idea of DT concept in early days was simply a hyper-realistic simulation of a physical process. As to achieve it, data stream continuously fed into the simulation to keep it up-to-date in operational time. Interaction is restricted to physical-virtual and virtual-virtual. This concept quickly grew obsolete as researchers recognize innovation in information technology allows supplementary knowledge, which in turn can be applied to enhance the physical process. Literature refers to them as narrow sense and broader sense of DT. Some literature have gone beyond and conceptualized the mirror simulation into digital shadow (DS) in contrast with DT [2]. DT is often modeled as three dimensions:

- Physical space covers physical devices, such as the object of DT and sensors.
- Virtual space is the digital representation of the physical space
- Information layer which is responsible for connection and data processing exchange between two spaces.

Tao et at. [3] suggests a complete DT where information layer is split into three dimensions

- Connection dimension is mainly responsible for data exchange between components.
- Data dimension is responsible for data storage.
- Service dimension provides data-driven information than can enhance the systems.

Their five-dimensional model put an emphasis on the difference between DS and DT. The interaction between virtual-physical is added forming a control-feedback loop. In many application of DT, the loop is semi-automated with human-in-the-loop. Instructions and feedback are given to operators who will make changes to physical systems.

The rest of the paper is structured as follow: Section II introduces the Arrowhead Framework. Section III describes the DT

project. Section IV show the structure and implementation of our experimental laboratory with replicate physical twin. We discuss the result in Section V, related works in Section VI. We conclude the result in Section VII.

II. Arrowhead Frameworks

Arrowhead Framework is a Service Oriented Architecture (SOA) framework that supports interoperability in systems of systems including IoT devices. The framework use the local cloud concept which allows large scale of IoT components to be organized into several local clouds based on the geographical location of devices and communicate through inter-cloud service exchange [4]. The framework has its own set of terminology, in which two important definitions are quoted from Varga et al. [5]

> A Service is what is used to exchange information from a providing System to consuming System.
> A System is what is providing and/or consuming services

An Arrowhead local cloud comprises two parts: application systems that run main functionalities of the local cloud, and core systems coordinating application systems. Three mandatory core services are required for a functional local cloud (Figure 2).

- Service Registry stores information of all services and enables discovery of services
- Orchestration provides information of service consumption
- Authorization issues Authentication and Authorization within local cloud

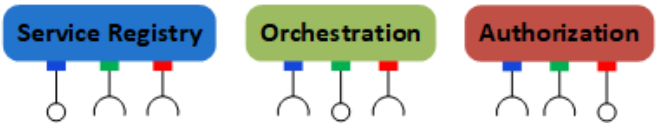


Fig. 1: Three mandatory core systems of Arrowhead Framework. Blue, green, and red rectangles are commonly used to denote the connection to the three mandatory core systems of Arrowhead Framework in related literature

.

In addition, there are several supporting core systems to improve quality of services. Gatekeeper and Gateway are responsible for inter-cloud communication. Eventhandler offers data exchange model in the publish-subsribe manner similar to event-driven model. However at the implementation core, Eventhandler follows service oriented architecture [6]. Publisher and subscriber communicate with Eventhandler through discover-binding. Aziz et al. [6] did a comparison between Eventhandler and Message Queuing Telemetry Transport (MQTT), a prominent protocol using event-driven model. They showed that the Eventhandler has slightly higher latency but still adequate. However Eventhandler offers extra benefits in orchestration support, security, stability, availability and failover. They also suggest Eventhandler would provide better support for DT.

The concept Autonomic Computing, systems that govern itself was first introduced into Arrowhead Framework by the works of Lam et al. [7], [8] to solve interoperability problems. The idea is to create a monitoring loop that continuously gathers information of its local cloud and devises adaptation plan to application systems when it detects a violation of the local cloud's policy. Their works combine IBM's adaptation control loop MAPE-K with semantic web technologies into a supporting core system called Autonomic Orchestration. It is later renamed as Autonomic Adaptation System (AAS) to avoid confusion with the Orchestration core system. The AAS does not collect information directly from application system, but utilizing information from other supporting core system. Monitor-block communicates Data Manager and Orchestration core systems to collect information. Analyze-block acts as semantic extractor which infers knowledge from raw data. Plan-block works out adaptation plans based on input adaptation polices from users. Execute-block delegates adaptation plans to other core systems: Configuration core system carries out re-configuration actions, Orchestration core system conducts re-orchestration actions. Knowledge-block stores knowledge in the form of ontology triples.

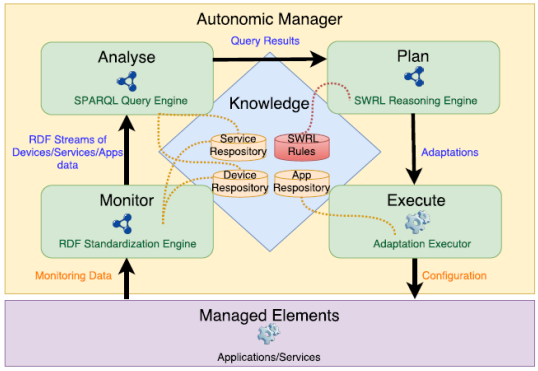


Fig. 2: The conceptualize adaptation loop [7]

.

III. Norwegian Uses Case

The Norwegian Use case is a part of ECSEL project The Arrowhead Tools for Engineering of Digitalisation Solutions. Its objective is to build a complete solution for DT with main focus on engineering infrastructure [9]. The physical object is the Palfinger crane mounted on NTNU research Gunnerus vessel [10]. There are four partners working together in the project: NTNU Ålesund[1], Jotne[2], Tellu[3] and HIOF[4]. Each partner has different interest and is responsible in different aspects of the use case, which makes the use case preventative of common business projects with multi-stakeholders. The roles in DT can be roughly distributed in accordance with the five

[1]https://org.ntnu.no/intelligentsystemslab/
[2]https://jotne.com/
[3]https://tellu.no/en/
[4]https://www.hiof.no/iio/itk/english/research/groups/cyber-physical-systems/

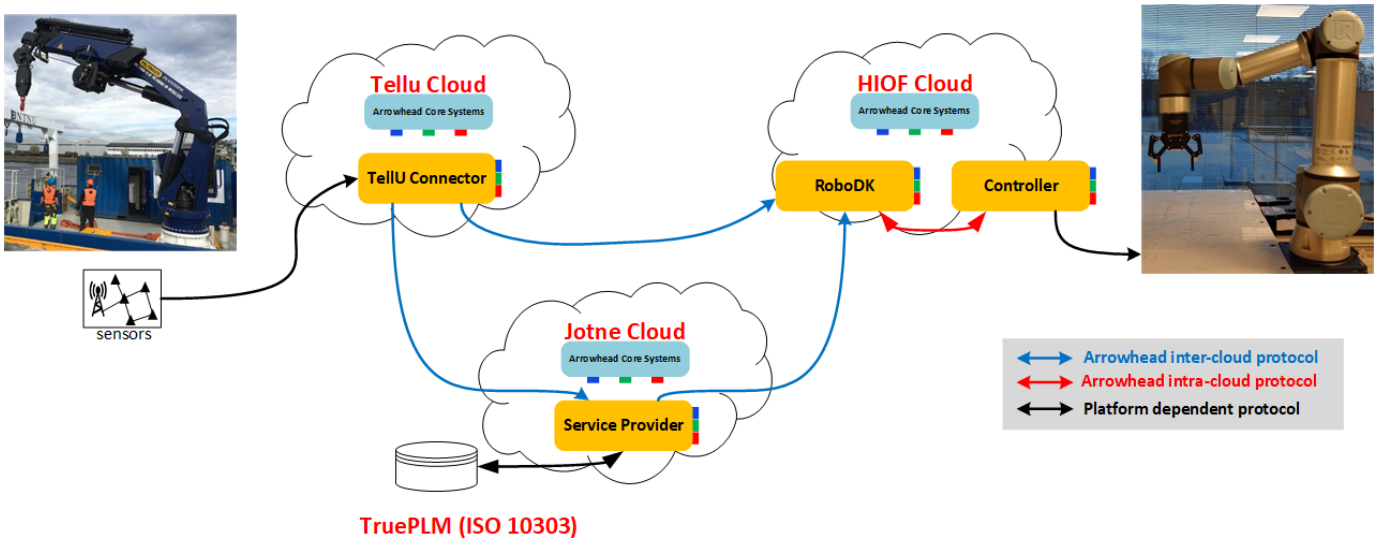


Fig. 3: Local cloud architecture of the use case. Tellu Cloud creates connection to sensor data, the TelluConnector service is the service endpoint that allows other service to access this data. Jotne Cloud provides data storage standardized with ISO 10303. HIOF Cloud is our experimental laboratory. It has two application systems: mock-up DT with RoboDK, and a connector application system responsible for interacting with URe16.

dimensional model of DT. Tellu is in charge of edge devices, sensor investigation and installation which belongs to the physical dimension. Jotne develop an ISO 10303 repository for DT data (data dimension). ISO 10303 [13] provides common exchange formats as well as common semantic for the data that can be reused or shared between different entities. NTNU Ålesund research is on the virtual and service dimension. They have developed a high fidelity simulation of the crane and the ship it is mounted on [11]. Based on that, many information services can be driven from their DT [12]. Our interest is on how to maintain the synergy of systems of systems (connection dimension). It has two facets: interoperability of components in a DT, and adaptation to system evolution. Experimenting on either the real crane or NTNU's DT is impractical for logistic and security reasons. We have built an indoor laboratory for experimenting using our URe16 robot to replicate the crane. Overview of data flow from the crane to our laboratory is illustrated in Figure 3.

IV. Laboratory

A. Feedback-control-loop

Adding the URe16 completes the feedback-control-loop as illustrated by Figure 4. *Data Collection* encapsulates information retrieved from the physical space, i.e. the sensor data installed on the crane. In addition, data that can be collected from URe16 includes joint positions and joint velocity. *Data Collection* includes data from crane's sensors and data from the replicate physical twin. *Digital Twin* is RoboDK's simulation of URe16. Knowledge from *Digital Twin* is communicated back to the *Physical twin* which is URe16, via *Controller*. Information exchange in RoboDK and *Controller* is abstracted into Arrowhead service and interacting via Arrowhead Protocols. There are two data channels: *offline*

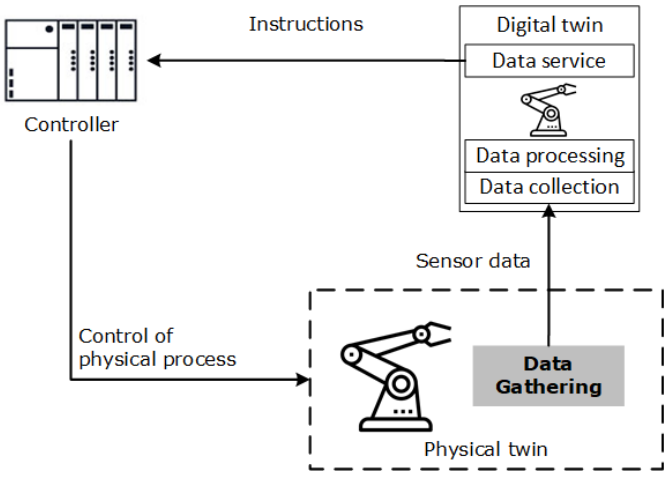
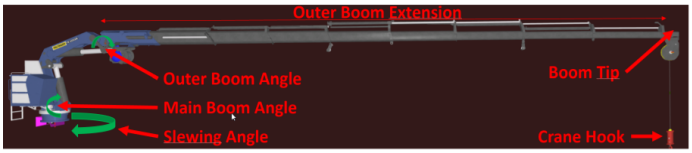


Fig. 4: Feedback-control-loop in the experimental laboratory.

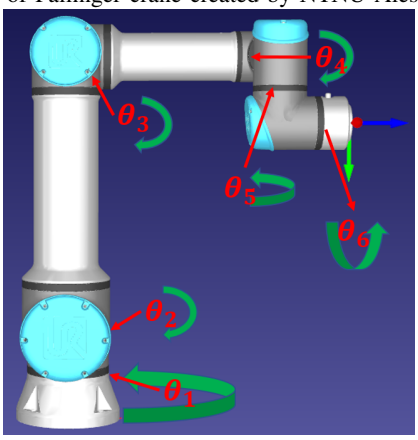
.

data fed from Jotne's data storage and *online* data fed from Tellu Cloud

1) Offline data of crane is fed from Jotne's data storage. This is useful for analysis historical data. In addition, the crane is not always in operation. There is only a small fracture of data from historical data is meaningful for analysis. The channel is not continuous, channel is establish via *find-bind*. RoboDK consumes service $trueplm - get - sensor - data - service$ from Jotne Cloud.



(a) DT of Palfinger crane created by NTNU Ålesund [12]



(b) Mock-up DT with RoboDK's URe16 model

Fig. 5: Mapping between two 3D models

2) Online data from crane is streamed directly from Tellu Cloud. Communication between RoboDK and URe16 is also an online data channel. Continuity is a requirement for online data. Arrowhead framework provides Eventhandler system that provides the publisher-subscriber service which can fulfill this requirement.

B. Replicate of crane with URe16

The collaborative industrial robot URe16 is chosen as the replicate physical twin because of its similar anatomy with the Palfinger crane that is enough to replicate movements of the crane. URe16 has six joints $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ which are shown in Figure 5 (b). Sensor data from crane is a set of six measurements ($slewing\_angle$, $main\_boom\_angle$, $outer\_boom\_angle$, $outer\_boom\_length$, $differential\_pressure$, $error\_bits$). We only map a subset of sensor measurements to a subset of joints due to the different characteristics of the crane and URe16, while retaining enough information of crane's activities of our interest.

- θ_1 represent the $slewing\_angle$ of the crane.
- θ_2 is mapped with the $main\_boom\_angle$ of the crane.
- θ_3 corresponds to the $outer\_boom\_angle$ of the crane.

The rotation orientation of each joint is illustrated with the green arrows. All joints have rotation range from -360 to 360. However to avoid collision and to replicate the crane's activities, their rotation range are limited to the list in Table I.

C. Digital Twin

The structure of our DT is illustrated in Figure 4. *Data collection* comprises Arrowhead Services that use the protocols from Arrowhead Framework to communicate with data sources. It outputs ($slewing\_angle$, $main\_boom\_angle$, $outer\_boom\_angle$, $outer\_boom\_length$, $differential\_pressure$, $error\_bits$).

Sensor	s_{min}	s_{max}	Joint	θ_{min}	θ_{min}
Slewing angle	-4700	-1800	θ_1	0	180
Main boom angle	-46	75	θ_2	220	270
Out boom angle	1200	1500	θ_3	210	290

TABLE I: Joint limitation and parameters for Equation 1

Data processing function $f(s)$ transform raw data into meaningful data for the DT. It can include time series techniques for denoising or feature extractions. In our experiment, the $f(s)$ takes a subset of input and transforms it into URe16's joints: $s(slewing\_angle, main\_boom\_angle, outer\_boom\_angle) \rightarrow \theta(\theta_1, \theta_2, \theta_3)$. The process includes two steps: a min-max scale is applied to fit the sensor measurement into limitation ranged defined for URe16 (Equation 1). The parameters for this equation are listed in Table I.. The second step is to apply a filter to remove the spikes in time series by checking if the measurement shoot up in a very short time $\Delta\theta > \gamma$ with γ is set at 100.

$$\begin{aligned} \theta &= f(s) \\ &= \frac{(s - s_{min}) * (\theta_{max} - \theta_{min})}{s_{max} - s_{min}} + \theta_{min} \end{aligned} \tag{1}$$

Figure 6 shows the comparison between input and output data from *Data processing*. The spikes from data are remove. Data is scaled to appropriate scale for URe16 while retaining the movement patterns of interest.

Fig. 6: Data comparison between before and after data processing

.

Virtual representation of the DT in choice is the URe16 model come in RoboDK (Figure 5 (b)). It is the virtual representation of the replicate physical twin which is URe16.

Data service function g responsible for providing useful knowledge produced by DT and is communicated to the physical twin. For our experiment, we want to test the accuracy of our DT compared to the crane on the replicate physical twin. The instructions for URe16 is the joint positions p, which is a VECTOR6D of for the angles of six joints. The data service

function $g(\theta)$ simply takes input θ and filled in two zeros for p (Equation 2).

$$p = g(\theta) = (\theta_1, \theta_2, \theta_3, 0, 0) \tag{2}$$

Data service function g can be machine learning algorithms or algorithms to regulate the replicate physical twin, e.g. controlling the joint movement speed based on the workload of the tip of the crane.

D. Adaptation to dynamicity

To satisfy the constraint that the Outer boom will not move faster than the speed limit. The joint velocity from URe16 can be fed directly to RoboDk. After comparing with the speed limit, RoboDk will configure itself to reduce the frequency of its output instructions. This approach has many shortcomings, as it difficult to manage changes in the system in the long run due to system maintenance and system evolution. For example, speed limit of the robot can be increased during summer or holiday when the lab has less visitors and vice versa. The developers need to manually adjust the configuration of the robot. An alternative approach to handle this situation in a separation of concern manner, by leaving the logic that is highly dependent on the dynamicity to be handled by AAS from Arrowhead Framework [8]. Data flow is re-routed as shown in Figure 7. Joint velocity is streamed into Data Manager, which can be accessed by AAS. Operator can provide adaptation policies to AAS using semantic language, which is shown in Figure 8. AAS constantly monitors speed measurement and informs RoboDK when the speed limit is violated. Adaptation policies are tied to semantic extraction which can be manipulated by users. *:OuterBoomSpeed* and *:OuterBoomSpeedLimit* are subjected to semantic extraction. Users can assign a number to *:OuterBoomSpeedLimit*. The raw joint velocity fed from URe16 is a VECTOR6D for six joints $(v_1, v_2, v_3, v_4, v_5, v_6)$. User can identify *:OuterBoomSpeed* as v_3. In scenarios where a different model of robot arm is used, the adaptation policy remains the same while *:OuterBoomSpeed* is associated with a different measurement. Both adaptation policy and semantic extraction can be performed at run-time without interrupting the operations of the systems.

V. Discussion

The experimental laboratory address problems of connection and security in performing the feedback-control-loop between digital and physical twin by using a replicate physical twin. The setup infrastructure allows elaborate experiment regarding changes in the environment while keeping the original systems intact. The alternative approach of using monitoring-adaptation provided by AAS from Arrowhead framework allows objectives of the systems be handled separately from component logic. It enables smooth inteoprability of devices. The adaptation policy in Figure 8 is applicable in either case when DT unit or physical unit is replaced. AAS keeps track with the data stream from physical unit and informs DT of adaptation policy to reconfigure in accordance of new components. The language used to describe the policy is semantic web technology, which

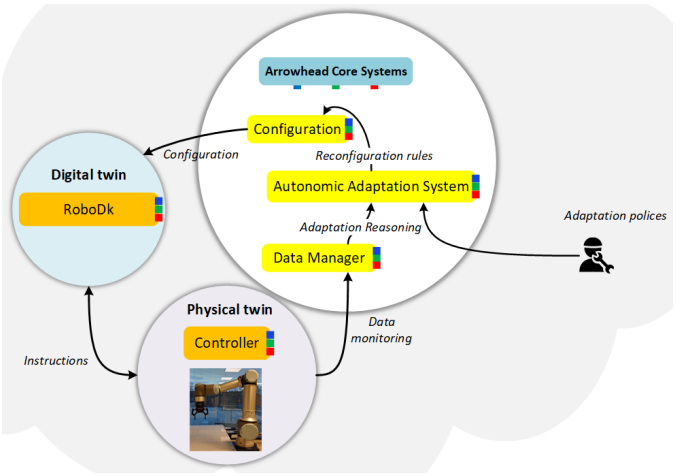


Fig. 7: Feedback-control-loop with participation of multiple Arrowhead core systems

.

```
(?obs sosa:mabeBySampler ?gadget)
(?obs sosa:hasSimpleResult ?result)
(?obs sosa:hasFeatureOfInterest :OuterBoomSpeed)
greaterThan(?result, :OuterBoomSpeedLimit)

(?gadget :hasService ?subscriber)
(?subscriber rdf:type :Subscriber)
(?subscriber :subscribeToEvent ?event)

(?p rdf:type :Publisher)
(?p :publishToEvent ?event)

(?event :hasName :MoveJoints)
-> configure(?p, "Outer Boom Speed",
                 :OuterBoomSpeedLimit)
```

Fig. 8: Adaptation policy for configuring outer boom speed with respect to the speed limit

can be difficult for users to use but is suitable for machine. Our future work aim to develop an automated solution to generate adaptation policies automatically based on historical data and feedback from environment. We speculate as systems evolve, their objective should remain the same, thus automatic adaption needs to retain the objective of the system. We will use this infrastructure to experiment system evolution with regard to the objectives of the systems.

VI. Related Works

Communication between virtual-physical space can be roughly divided into two approach. The first approach with human-in-the-loop. DT provides information and instruction to support decision making, and leaves the action to human operators. Laaki et al. [14] built DT of robotic arm in virtual reality (VR). Users used VR to remote control the robotic arm to perform surgery on test subject. Major et al. [12] built a control centre to remote control the crane and its vessel. Human operator controlled via a crane joystick. DT in both cases provide information regarding the feedback from

physical twin, e.g. position, orientation. The second approach is to output to physical twin directly. Hu et al. [15] and Liu et al. [16] both used printer as an example system to validate their connecting framework for DT. They used MTConnect and MQTT respectively in their works. Instructions were sent as notification messages, triggering the printer actions.

VII. Conclusion

We presented an experimental laboratory that allows testing DT logic on physical space using a replicate of physical twin. The laboratory shows a feedback-control-loop that allows instruction to enter the physical space. The infrastructure provides a safe space for experiments on replicate physical twin without risk on the original physical twin. We also demonstrate using adaptation monitor loop from Arrowhead core system AAS in addition to the feedback-control-loop that can enhance the adaptivity in run-time.

Acknowledgement

This research is funded by ECSEL, the Electronic Components and Systems for European Leadership Joint Undertaking under grant agreement No 826452 (Arrowhead Tools), supported by the European Union Horizon 2020 Research and Innovation Programme and by the member states.

References

[1] Piascik, R., et al., "Techonology Area 12: Materials, Structures, Mechanical Systems, and Manufacturing Road Map", pp. 12, 2011.

[2] Bergs, T., Gierlings, S., Auerbach, T., Klink, A., Schraknepper, D., Augspurger, T., "The Concept of Digital Twin and Digital Shadow in Manufacturing". In *Procedia CIRP*, vol. 101, pp. 81-84, 2021.

[3] Tao, F., Zhang, H., Liu, A., Nee, A., "Digital Twin in Industry: State-of-the-Art". In *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405-2415, 2019.

[4] Delsing, J., Varga, P., "Local automation clouds". In *IoT Automation: Arrowhead Framework*, pp. 27-42, 2017.

[5] Varga, P., Blomstedt, F., Ferreira, L., Eliasson, J., Johansson, M., Delsing, F., de Soria, I., "Making system of systems interoperable-the core components of the arrowhead framework". In *Journal of Network and Computer Applications*, vol.81, pp.85-95, 2017.

[6] Aziz, A., Scheln, O., Bodin, U., "Data Integration Models for Heterogeneous Industrial Systems: A Conceptual Analysis". In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1-8, 2021.

[7] Lam, A., Haugen, Ø., "Supporting IoT Semantic Interoperability with Autonomic Computing". In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pp. 761-767, 2018.

[8] Lam, A., Haugen, Ø., "Applying semantics into Service-oriented IoT Framework", in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, pp. 206-213, 2019.

[9] Intelligent Systems Lab, *EU H2020: Arrowhead Tools for Engineering of Digitalisation Solutions*, Ålesund, Norway. Accessed on: Jan. 20, 2022. [Online]. Available: https://org.ntnu.no/intelligentsystemslab/project/euh2020.html

[10] Gunnerus Research Vessel - NTNU, *Gunnerus Research Vessel*, Ålesund, Norway. Accessed on: Jan. 20, 2022. [Online]. Available: https://www.ntnu.edu/gunnerus

[11] Lozano, C., "Digital Twin for Structural Monitoring and Predictive Maintenance of a Marine Crane", M. S. thesis, Faculty of Engineering, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, June 2020. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2782170

[12] Major, P., Li, G., Zhang, H., Hildre, H.P., "Real-Time Digital Twin Of Research Vessel For Remote Monitoring". In *ECMS 2021 Proceedings*, pp.159–164, 2021.

[13] Pratt, M., "Introduction to ISO 10303 - the STEP Standard for Product Data Exchange". In *Journal of Computing and Information Science in Engineering*, vol.1, pp. 102-103, 2001.

[14] Laaki, H., Miche, Y., Tammi, K., "Prototyping a Digital Twin for Real Time Remote Control Over Mobile Networks: Application of Remote Surgery". In *IEEE Access*, vol. 7, pp. 20325-20336, 2019.

[15] Hu, L., Nguyen, N., Tao, W., Leu, M., Liu, X., Shahriar, R., Sunny, S M N., "Modeling of Cloud-Based Digital Twin for Smart Manufacturing with MTConnect". In *Procedia Manufacturing*, vol. 26, pp. 1193-1203, 2018.

[16] Liu, C., Jiang, P., Jiang, W., "Web-based digital twin modeling and remote control of cyber-physical production systems". In *Robotics and Computer Integrated Manufacturing*, vol. 64, 2020.