

Received April 28, 2022, accepted May 23, 2022, date of publication May 27, 2022, date of current version June 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3178424

A Methodology for Generating a Digital Twin for Process Industry: A Case Study of a Fiber Processing Pilot Plant

MOHAMMAD AZANGOO¹, (Member, IEEE), LOTTA SORSAMÄKI²,
SEPPO A. SIERLA¹, (Senior Member, IEEE), TEEMU MÄTÄSNIEMI², MIIA RANTALA³,
KARI RAINIO², AND VALERIY VYATKIN^{1,4}, (Fellow, IEEE)

¹Department of Electrical Engineering and Automation, Aalto University, 00076 Helsinki, Finland

²VTT Technical Research Center of Finland, 02044 Espoo, Finland

³Semantum Oy, 02150 Espoo, Finland

⁴Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 971 87 Luleå, Sweden

Corresponding author: Mohammad Azangoo (mohammad.azangoo@aalto.fi)

This work was supported by Business Finland under Grant 3915/31/2019 and Grant 4153/31/2019.

ABSTRACT Digital twins are now one of the top trends in Industry 4.0, and many companies are using them to increase their level of digitalization, and, as a result, their productivity and reliability. However, the development of digital twins is difficult, expensive, and time consuming. This article proposes a semi-automated methodology to generate digital twins for process plants by extracting process data from engineering documents using text and image processing techniques. The extracted information is used to build an intermediate graph model, which serves as a starting point for generating a model in a simulation software. The translation of a graph-based model into a simulation software environment necessitates the use of simulator-specific mapping rules. This paper describes an approach for generating a digital twin based on a steady state simulation model, using a Piping and Instrumentation Diagram (P&ID) as the main source of information. The steady state modeling paradigm is especially suitable for use cases involving retrofits for an operational process plant, also known as a brownfield plant. A methodology and toolchain is proposed, consisting of manual, semi-automated and fully automated steps. A pilot scale brownfield fiber processing plant was used as a case study to demonstrate our proposed methodology and toolchain, and to identify and address issues that may not occur in laboratory scale case studies. The article concludes with an evaluation of unresolved concerns and future research topics for the automated development of a digital twin for a brownfield process system.

INDEX TERMS Digital twin, process industry, modeling, steady state simulation, image recognition, text recognition, directed graph, piping and instrumentation diagram, flowsheet population.

I. INTRODUCTION

The Fourth Industrial Revolution, also known as Industry 4.0, is characterized by rapid technological changes in response to new industrial requirements, such as more flexible inter-connection, agile and smart automation systems, and big data handling. Chemical, pulp and paper, heat and power industries are examples of process industries that use continuous manufacturing or use indistinguishable batches of materials to manufacture their products. Process industry, like

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Merlino¹.

other industrial sectors, must adapt to Industry 4.0 compatible technologies to speed decision-making, optimize processes, reduce risk and remain competitive and agile. As a result of technologies such as digital twins, the boundaries between the physical and digital worlds are becoming increasingly blurred. Digital twins use a model of an industrial plant to simulate the plant's behavior and assist the process by interacting with it.

There are many different simulation-based approaches under the umbrella of Industry 4.0 like Virtual Commissioning (VC) and Agent-Based Modelling and Simulation (ABMS) [1]. A key development in simulation technology

during the fourth industrial revolution is the digital twin concept [2]. In the context of the process industry, a digital twin can be defined as a virtual representation of a physical system that is integrated into the actual process data. Kritzinger *et al.* [3] defined, based on the level of integration, the digital model as a digital representation of a physical system with no automated data exchange, a digital shadow as a system having one-way data exchange, and finally a digital twin as a system having fully integrated, automated two-way data exchange between the physical and digital system. The digital twin can use existing communication infrastructure, such as a process automation network, to communicate with the physical system in a two-way manner [4]. A digital twin is a set of virtual objects that can simulate the behavior of the actual system in the deployed environment, and includes the features, condition and behavior of the real-life system through accessible virtual assets, like models and data [5]. They can be used for different applications and purposes including more responsive and efficient product design, optimization of the system, the digitalization of production facilities and development of more accurate control systems [6].

To create a first-principles digital twin, a digital model is first imported into a simulation software and then integrated into the actual process data. Due to unknown or unpredictable prices, IT expertise requirements, and lack of simulation tool knowledge, SMEs use simulation models and digital twins in non-standard formats for production systems [7]. Using software and tools can speed up the digital twin generation process. Using image and text recognition software can automatically extract data from engineering documents, if they are not in an Industry 4.0 format. Processing extracted data based on semantics and logic can help to build a process simulation model and digital twin.

The approach towards the generation of a digital twin for brownfield process systems presented in this study was adopted from the general road map presented by Sierla *et al.* [8]. Brownfield process systems are functioning plants that may have been developed and built before modern digital systems and may thus lack design information in an Industry 4.0 format. A preliminary study on the semi-automatic generation of a digital twin for a laboratory scale water process plant [9] was recently published. However, it was limited to a laboratory scale process plant with limited types of process equipment. It involved manual work for component selection, and assumed an available machine-readable P&ID file.

This paper discusses a variety of approaches and software tools that have been developed independently to aid the creation of digital twins for brownfield process systems. The proposed approach continues the earlier work and presents the scale-up process for the semi-automatic generation of a digital twin. A pilot scale fiber process is used as a case study. The case is much more complicated than the laboratory scale works in previous case studies. The main contribution of this work is to design, develop and implement a methodology and toolchain for the semi-automatic generation of digital twins and to test it with this case study.

Figure 1 shows the general structure of the proposed approach and the interconnections between the different software and tools used. This approach extracts required information from Piping and Instrumentation Diagrams (P&IDs), creates a machine-readable model of the system automatically, and lists semi-automatically the specific process components and pipeline connections wanted to be included in the digital model of the system. The ability to extract essential inputs from a P&ID using two distinct solutions with varying levels of fidelity and certainty allows for a lot of flexibility in dealing with diverse use cases. Then an intermediate graph-based model is built, containing information about different process components, pipeline connections and attributes related to components like position, rotation, type and source file. The intermediate graph model will be adapted for steady state simulation, which requires information such as ports of process components, which is not specified in the P&ID. A steady state is a state of equilibrium in which all state variables in a process system remain constant. Finally, to be able to call the simulation model as a digital twin, it must be integrated into the actual process data with an on-line or off-line connection. However, this last step is not covered in this paper. The contribution of each specified program and tool to methodology formulation and development, as well as their structure and format of inputs and outputs, will be discussed in this study.

This paper is structured as follows. Section II will review the existing literature. Section III will present a generic methodology to create steady state digital twins for brownfield process systems. Section IV will present a brownfield fiber processing plant as a pilot scale case study. Results of the implementation of the proposed methodology at the case study site will be provided and discussed in Section V. The last section concludes the paper by outlining potential next actions for improving and expanding the scope of this work. Additionally, there are two appendices about the positioning of components for visualization and simulator-specific mapping rules for acting as an interpreter between the P&ID and the simulator.

II. LITERATURE REVIEW

Simulation-based research in Industry 4.0 is on the rise [1]. The digital twin is the simulation model development flagship in Industry 4.0. For example, Vijayakumar *et al.* [10] proposed replacing human operation of a manufacturing facility with a digital twin in order to keep the model updated and simulated in real time, as well as to reduce the cost and time necessary for operation. Theoretically, digital twins are presently at the rapid growth stage [6]. It indicates that researchers and industry are gradually accepting digital twins. Making digital twins isn't straightforward. Essential aspects of the digital twin for brownfield plants need to be defined. Bamberg *et al.* [11] tried to list the essential items and benefits of a digital twin from the user's perspectives; to ensure the digital transformation, a list of questions relating to requirements and probable challenges should be reviewed.

However, the ultimate answer varies based on the system's aims, structure, and domain. A repeatable infrastructure will speed up and automate the generation of digital twins from available engineering documents and help to track changes during the life cycle [12].

A. PROCESS SIMULATION; CORE OF A DIGITAL TWIN

Digital models, including steady state or dynamic first-principles process simulation models, data-driven models and combinations of these two, i.e. hybrid models, constitute the core of a digital twin [4], [13]. A steady state simulation, which will be considered in this paper, is used to look into a system's behavior when it is operated without disturbances, operator input or other transients. The main difference between steady state and dynamic simulation is that steady state assumes that variables are consistent across time and there is no accumulation in the system, so the overall mass and energy are fixed. To find the best design parameters and operating conditions for systems in a short time, steady state simulation is used. Rosen *et al.* [14] see the digital twin concept as the next wave in process modeling and simulation emerging after simulation-based system design and engineering. Martinez *et al.* [15] present a tracking simulation architecture for process systems using the data history of the process to update the first-principles model and keep the tracking model updated during the system's life cycle.

Though a digital twin is similar to a digital process simulation model, it is much more. Process simulation models focus on what could happen in the real world, but not what is currently happening, whereas digital twins can be used for monitoring, controlling, diagnosing and predicting the current state of the process [16]. To enable this, the digital twin must integrate the process model into the current process data [4].

Computer-based simulation of processes using first-principles models dates back to the 1980s, and has since then been widely used as a design and modeling tool for various industries [17]. Sorsamäki *et al.* [18] reviewed some of the scientific literature on using both steady state and dynamic process simulation in pulp and paper applications. However, none of these digital models have fully integrated automated data exchange between the physical system and its digital counterpart, and thus they don't meet Kritzinger's definition for digital twin [3].

B. DIGITAL TWINS IN PROCESS INDUSTRY

The digital twin term originates from the aerospace industry in early 2000s [19], and has been used since early 2010s in manufacturing industry [20]. However, it is only in recent years that process industry have adopted its usage; chemical industry [13] and pulp and paper industry [21] in the front line, while food processing industry [4] and biomanufacturing industry [22] are still falling behind other process industries in terms of its implementation.

It is neither simple nor straightforward to use digital twins in industry. Two years ago, ARC Advisory Group published

a white paper about the prerequisites for using digital twins in process industry [23]. The white paper stated that an organization is digitally ready when it has reached a minimum level of digital maturity in resources, systems, organization and culture. This maturity level of the company is the key character to guarantee the success of digital transformation and the value of the digital twin. The implementation of digital twins will make feasible the transition to smart processing characterized by a high level of automation due to the extended use of remote sensing, real-time data acquisition and monitoring, and advanced visualization tools [4].

Examples of the usage of digital twins in process industries exist. Industry 4.0 is profoundly affecting the digitalization need in pulp and paper industry. Andritz has responded to this by creating a digital twin application that combines the simulation software IDEAS with an execution platform where human interactions can be implemented into continuous processes [24]. Carlberg [21] presented an interesting vision of an autonomous pulp mill of the future as a mill that benefits from the use of digital twins utilizing a dynamic process simulation model coupled with a control model of the real-time control system to allow the autonomous mill to "run itself" with little or no human intervention. Örs *et al.* [13] suggested a generic framework for AI assisted digital twin in chemical process industry from an operational perspective. The main focus in their paper was, however, on the conceptual formulation, and further practical implementation will be needed to validate the framework. Koulouris *et al.* [4] presented the methodology for the application of integrated process model and digital twin model aiming to enhance the production planning and scheduling of an industrial scale beer production and filling facility. Udugama *et al.* [22] presented a framework built upon a five-stage pathway starting from a basic steady state process model and ending to a fully-fledged digital twin for a second-generation ethanol fermentation process. Digital twins can be used as a control strategy development tool to enable the development of optimized controllers which can increase the efficiency of bioprocess systems while they are in normal operation [25].

C. OBJECT RECOGNITION FOR INFORMATION EXTRACTION FROM ENGINEERING DOCUMENTS

Using computer vision tools can speed up and improve the quality of digital asset generation. In fact, most industrial process automation relies on these techniques. Videos, 3D models, point clouds, drawings, engineering documents, check lists, and operational data are examples of source data.

Computer vision techniques can see and detect objects, compare and categorize them using database information. For more complex tasks like segmentation, scene understanding, object tracking, image captioning and event detection, advanced computer vision techniques can be used [26]. Major software providers have each their own optical character recognition (OCR) packages. Nowadays, these products are mature and commercially available as software libraries. There are a few Open-Source software libraries available for

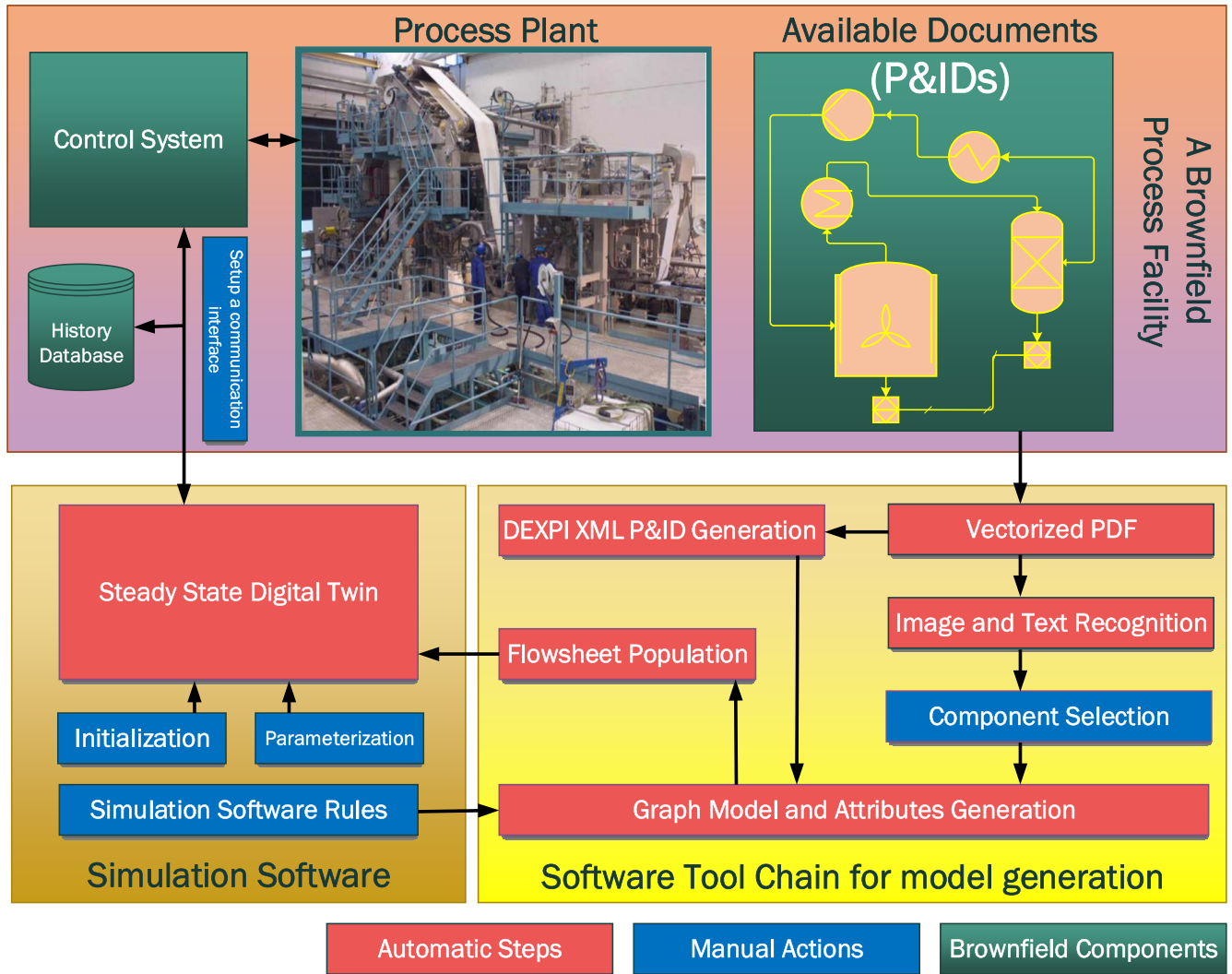


FIGURE 1. Structure and connection between subsystems for steady state digital twins.

OCR, and of these Tesseract is generally considered the best [27]. Tesseract version 5.0 was used in the presented work to extract wanted text snippets in a P&ID.

Many machine vision tools have been developed for dealing with engineering documents such as P&IDs. Yu *et al.* [28] use deep learning to recognize data from P&IDs. Preprocessing the diagram involves removing borders and realigning it. This stage uses an AlexNet deep learning model to recognize symbols, and a connectionist text proposal network (CTPN) to recognize text. Also, image processing is used to recognize piping, signal lines, and tables. Ali *et al.* published another study on deep learning algorithms for text and symbol detection of P&IDs [29]. In this method, OpenCV library detects geometrically structured objects after identifying text in diagrams. According to Kang *et al.* [30], a database of registered symbols is used to recognize symbols from P&IDs using template matching. A sliding window method considers lines connected to recognized symbols. OCR recognizes text

and uses predefined attribute information for each symbol to connect them. Mani *et al.* [31] used a Convolutional Neural Network (CNN) to detect instrumentation symbols. Bounding boxes of text are recognized with Efficient and Accurate Scene Text Detector (EAST) and the text interpreted with Tesseract OCR. Connection detection is performed by transforming the pixels into a graph and performing a depth-first search. In the work presented by Rahul *et al.* [32], a fully convolutional network (FCN) is trained to recognize symbols and a connectionist text proposal network (CTPN) and Tesseract ORC are used to recognize text.

D. AUTOMATIC GENERATION OF DIGITAL TWINS

All of the examples mentioned in section II.B were based on manual generation of a digital twin. However, the main ambition is towards the automatic generation of digital twins.

Using available or extractable information of the brown-field process can be a good starting point for the automatic

digital twin generation. Digital twins for brownfield processes can be automatically extracted from available or extractable information such as 3D scan of production sites [32], [33], 3D models [35], P&ID documents [36], design phase requirements [37], archived data repository [38] and mixture of these information [39]. Sierla *et al.* [40] extracted graph models of process plants from two different sources, 3D CAD models and P&IDs, for the generation of digital twins. However, the extracted graph models were at varying levels of abstraction, making it difficult to compare them for validation purposes. Several algorithms have been proposed for converting a 3D CAD generated graph to make it at the same level of abstraction as a P&ID generated graph [41]. Siemens and Bentley Systems [39] collaborated to create an as-built digital twin of a brownfield process plant using a variety of data sources, including 1D (datasheets, lists, records), 2D (drawings, logical connectivity), and 3D (physical layout, sizes). They also considered data that was unstructured or unlabeled (dark data). Dark data can be made more visible by being tagged, validated, or linked to other data, resulting in more precise digital twins.

Automatic model generation is part of the story, but not the whole. To have a comprehensive digital twin generator, it is also necessary to implement and connect generation phases such as simulation software interface implementation, initial condition consideration, and equipment parameterization. To make this procedure more automatic, Sierla *et al.* [9] introduced several rules to convert an intermediate graph model into a format suitable for steady state simulation software. Azangoo *et al.* [42] demonstrated how machine learning can extract process parameters for digital twins from recorded process history.

There are limited research publications which describe a comprehensive chain of tools for extracting information from engineering legacy documents, processing intermediate extracted models, and transferring required info to a simulation model for a specific application. One of the most comprehensive works in this domain was presented by Arroyo *et al.* [43]. Their proposed methodology can extract model information from both raster images and vectorized graphic P&ID files and transfer them to Modelica modeling language. The generated model can be manually connected to a process control system to validate control system during Factory Acceptance Tests (FATs). Son *et al.* [44] present another general tool chain for the generation of updated model of the system from new point clouds, 3D models and P&ID files.

Solution for the automatic generation of digital twins is not limited to the process industry. Also in other domains, like manufacturing, several solutions have been suggested. To save cost, time and resources, and to make the process IT expert independent, Sommer *et al.* [45] developed some tools to automate part of the procedure for a digital twin generation of a manufacturing system. They used fast scans of the manufacturing site and subsequent object recognition in the point cloud to import the model of factory equipment into SolidWorks simulation software. For the

rest of the simulation model generation, human expert input is required. Liu *et al.* [46] presented an implemented and validated digital twin framework for metal additive manufacturing systems, where a cloud digital twin is connected to distributed edge digital twins in different product lifecycle stages. Stobrawa *et al.* [47] presented a semi-automatic solution for transferring extracted information from 3D scan using object recognition tools to a simulation software to create a virtual model of a production system for generating a digital twin. Their solution is exporting the object data to an XML-file and then transferring that to Plant Simulation.

E. FLOWSHEET IN ENGINEERING DESIGN

Process design and engineering phases produce flow-sheets which can be conceptual diagrams, process flow diagrams (PFDs) or piping and instrumentation diagrams (P&IDs). PFD and P&ID provide the functional design basis on which the rest of the system relies. Other required information and process engineering documents are often created and linked with these two most essential documents. P&ID contains detailed information about the process components, instruments, and control logic of the process. Since the application of P&ID documents is not limited to design phase but also operation and maintenance, it is the most common engineering document in the brownfield process systems.

The standard ISO 10303 provides a neutral and computer-interpretable mechanism for describing and exchanging product data independent from any particular system. Its application protocol AP231 [48] defines a process flow diagram as a schematic representation of process description (precisely process_definition element). The process definition is a set of process activities (e.g., unit operations) which transform or transport process materials to products or waste. These activities separate, mix, process or change physical state of materials and intermediate products and they are carried out by plant items or real major process equipment. A process topology organizes and orders process activities by hierarchical and stream relationships and specifies a boundary of process. The streams are flows into or out of unit operations and they are typically connected to ports of unit operations. A stream can be a material, energy, information, or signal stream. Each unit operation, stream and port have a symbol occurrence in the schematic representation. To generate the representation, the standard lists a lot of needed information which will be considered in the subsequent sections.

F. INDUSTRIAL STATE OF THE ART

Many commercial technologies claim to create digital twins automatically. In this field, there are a lot of competitors. But a closer look at these solutions reveals that most are still in development and require more time and effort to improve. The PIDgraph program [49] can build a DEXPI XML version of a PDF, DWG, and Bitmap P&ID file. This solution can't make a comprehensive digital twin right now, but a DEXPI file. Similarly, Model Broker software [50] can automatically extract data from old engineering documents, like P&IDs,

and convert it to a more open digital format, such as DEXPI XML files, allowing for easier validation and other use cases. UniversalPlantViewer [51] can automatically incorporate 3D models, laser scans of plants, P&IDs, isometric documents, and other engineering documents and drawings to improve collaboration and process plant life cycle optimization.

Siemens and Bentley Systems [39] developed a platform for evaluating unstructured and unreliable engineering data, as well as combining all available information from Excel files, P&IDs, and 3D models from various software packages into a single common asset data port for the generation of a more reliable visualized digital twin. Also, different digital twin-related software solutions will be able to combine and increase operational and asset performance and minimize downtime, damage, operational and IT risks, and information-related incidents and accidents through this alliance [52].

G. SUMMARY AND CURRENT WORK JUSTIFICATION

There are few research papers that explain a complete chain of tools for the digital twin generation. Based on the large diversity in the form of the models, simulation software, terminologies, and application in process engineering domain it is not possible to find or make a comprehensive and complete solution. However, development of specific software tools for different applications or use cases can pave the road for the future. Our proposed methodology is also made based on local requirements, needed level of fidelity, simulation type and environment. Steady state simulation can provide much information for plant owners with minimum efforts, so it can be considered as an affordable and fast solution in digital twin generation. We will discuss this in more detail in the following sections.

III. PROPOSED METHODOLOGY

In this section, we will present our methodology for the semi-automatic generation of a steady state digital twin for a brownfield process system. The general idea of this methodology is to use available P&ID engineering documents to extract and then process required information for the generation of the digital twin. The ideas for extension and improvement of this work will be discussed in section VI.

A. METHODOLOGY OVERVIEW

Figure 2 shows the proposed methodology for the semi-automatic generation of a digital twin for a brownfield process system. The figure presents a set of consecutive steps from initial engineering documents to digital twin implementation. As shown in the figure, we consider the P&ID document as the starting point (A1) of this work. Any form of P&ID (such as paper, scanned or DWG forms) can be converted to a more open PDF format (B1), from which objects and text can be extracted by the help of image and text recognition tools (B2). These tools help human experts to extract required process components from the P&ID to be included in the steady state simulation model of the system.

In our approach, the human expert makes a list of important process components and pipelines (B3) by selecting them manually with the help of image and text recognition tools (B2). To use the extracted data in a more efficient way, a software tool was developed to make an intermediate graph model from the generated lists of process components and pipelines (B4).

Alternatively, it is possible to convert a P&ID to a machine readable or digitalized P&ID format according to the standardized Proteus XML schema (C1), which can be considered an Industry 4.0 format. Then again, required information for the generation of intermediate graph model can be extracted by a developed software (C2).

Depending on the goals, simulation environment and required level of fidelity, the proper level of abstraction for the intermediate graph models was generated in steps B4 and C2. In the next step (D2), these two intermediate graph models are combined with simulation software specific input data (D1). In the presented work, the Balas steady state simulation software was selected as the use case software. Balas is a package for steady-state simulation of chemical processes, with a focus on pulp and paper. In our approach, the human expert defines manually mapping rules that identify the Balas specific simulation symbol(s) corresponding to the process component, which was selected from the P&ID. After that, a flowsheet populator software (D3) is used to populate the flowsheet of the simulation model into the user interface (D4). Finally, the steady state simulation model (D6) is achieved by manual initialization and parametrization (D5) of the flowsheet. According to Kritzinger's definition [3], the digital model (D6) converts to a digital twin (D8) when there is a two-way interconnection (D7) between the physical and digital system. After that, the human expert can run the steady state digital twin and use its results for improvement in efficiency and safety of the process plant.

It is important to consider the evolution of terminology in different steps and contexts to be clearer and avoid any confusion when reading this paper. A process component in an original P&ID file, for example, will be represented as a node in the graph model. Figure 3 depicts the general evolution, which is detailed further below.

In a process system, for a process equipment, such as a tank, the:

- Term **process component** is used in the context of P&IDs (A1, B1), DEXPI XML (C1) and text list which come out after using the Component_Selector (B2, B3)
- Term **node** is used in the context of graph model (C2, B4, D2)
- Term **unit operation** is used in the context of flowsheet population and simulator (D3-D8).

In a process system, for a pipe connection between two process equipment, the:

- Terms **pipeline, pipeline connection or connection** are used in the context of P&IDs (A1, B1), DEXPI XML (C1) and text list which come out after using the Component_Selector (B2, B3)

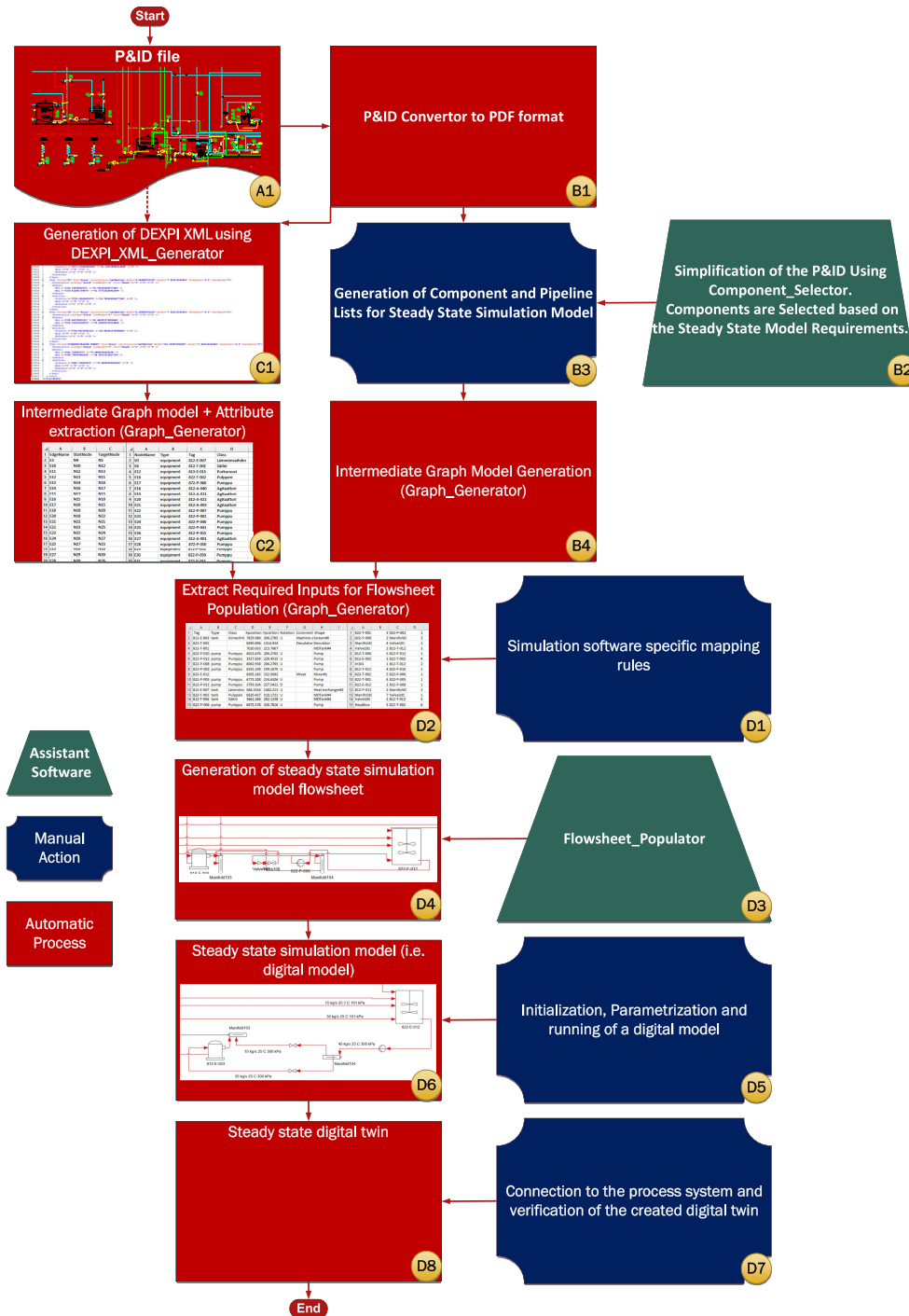


FIGURE 2. Proposed methodology for the semi-automatic generation of a steady state digital twin for a brownfield plant.

- Term **edge** is used in the context of graph model (C2, B4, D2)
- Term **stream** is used in the context of flowsheet population and simulator (D3-D8).

B. P&ID CONVERTOR (B1) AND COMPONENT_SELECTOR (B2)

The input for the Component_Selector tool is a P&ID in the PDF format. The PDF format can be either created

by scanning a paper version of the P&ID or printing an AutoCAD file of the P&ID into a PDF file. The Component_Selector program recognizes and extracts text fields in the scanned PDF file by using the open-source tool Tesseract 5.0 (the Python program library *pytesseract* is used). If the PDF file is a print of an AutoCAD file, then Python library PyPDF2 is used to extract the texts. In both cases, the P&ID is shown to the user with the extracted text fields surrounded by rectangles. Usually, the text fields tell the names or IDs of

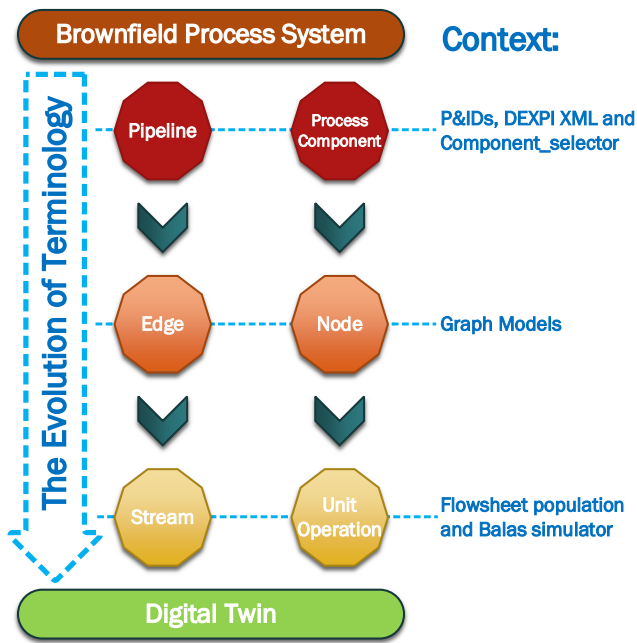


FIGURE 3. Terminology Evolution Path in this paper.

the process components in the P&ID. The view can also be zoomed in/out and scrolled.

C. GENERATION OF COMPONENT AND PIPELINE LISTS FOR A STEADY STATE SIMULATION MODEL (B3)

The aim of this step is to create a simplified subset of those process components and pipelines present in the P&ID that the human expert wants to be included into the steady state simulation model. It should be noted that a steady state simulation model is typically a simplified or reduced functional description of process component operations and pipelines, meaning that only those process components and pipelines are included that are required to solve the question posed to the model.

In our approach, the human expert uses the Component_Selector (B2) to select the required process components and pipelines. The selections are saved as text files.

D. DEXPI_XML GENERATOR (C1)

DEXPI_XML_Generator takes a vectorized PDF P&ID file as an input and translates it to a corresponding DEXPI file. This tool has been developed under the Model Broker product family as Model Broker for P&IDs [50]. As the graphics are already in a vectorized format, there is no need for image processing operations, such as line detection, that the approaches based on raster images might require. PDFs can also include texts as text objects, in which case text recognition step is not required. However, it is possible that the texts are represented with vectorized graphics instead of text objects, and *DEXPI_XML_Generator* can be configured to recognize the texts in these cases as well. The vectorized graphics of the

PDF are converted into a graph representation, from which the process component symbols and pipelines are recognized.

The translation is based on configurable presets, which contain the translation rules that define how the symbols, texts and connections are recognized from the PDF and mapped to DEXPI elements. Because the translation rules are configurable, *DEXPI_XML_Generator* can be used to translate diagrams that use different symbol sets and originate from different domains or design software.

The presets consist of patterns that describe the graphics and textual content of the symbols found in the PDF. To create a pattern, the user must select one instance of a symbol from the diagram. The user can then edit the graphics, mark the connection points to other elements or pipelines, and select the corresponding element in DEXPI. Texts related to the symbol, such as tag names or dimensional information, can also be included in the pattern. The text locations can be set both inside or outside the symbol and a corresponding attribute name can be given to each text. An example of configured pattern is shown in Figure 4. Once the pattern has been configured, it can be used to recognize other instances of the same symbol, including instances where the symbol has been rotated or scaled. The supported rotations include all 90-degree rotations and their horizontal flips. In comparison to other tools, finding only one instance of the symbol is enough to create the pattern for recognizing the symbol. There is no need to create training material by finding multiple instances of the same symbol.

When the configuration is ready, the translation can be performed to create the DEXPI file. Even if several different patterns match to the same PDF symbol, the application can choose the best pattern based on connectivity and the number of PDF graphical primitives that the patterns match to. The resulting file contains the recognized DEXPI elements, their attributes and their connectivity to other elements and pipelines. Each DEXPI element includes the location of the element and the graphics of the element that were extracted from the PDF. The unrecognized graphics are included under the Drawing element of the DEXPI file. The user can also create his own custom elements for the DEXPI file where the user can select the component type (Equipment, PipingComponent, ProcessInstrument etc), component class and generic attributes, and then map the PDF symbols to these DEXPI elements.

E. GRAPH PROCESSING (B4, C2)

The algorithms for steps B4, C2 and part of D2 are presented in this section. In these steps, input data from various sources will be used to create a graph model of the process system. Because graph modeling is both easy and flexible, it is commonly used to describe extracted models from P&ID files [52], [53]. In addition, under the umbrella of graph theory, there are numerous available algorithms, theories, and tools that can be utilized in the development, study and evaluation of the model. In graph models, process components are

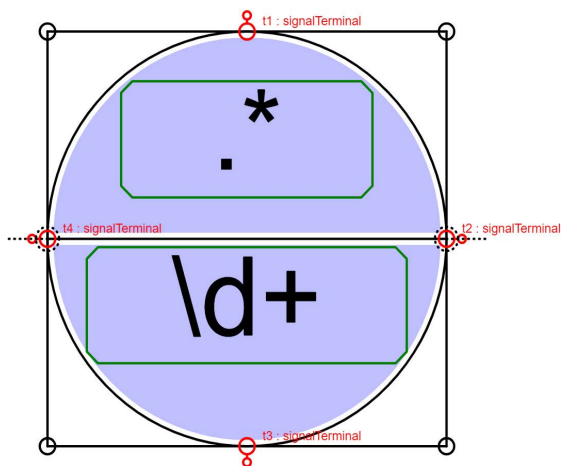


FIGURE 4. Example of a pattern for an instrumentation symbol. The graphics of the symbol have been converted into a graph. Four points marked in red show the connection points where a signal line can connect to the symbol. Two areas with darkened background indicate areas where text can be discovered and also contain a regular expression that the text should match.

known as nodes, while pipeline connections between them are referred to as edges.

Figure 5 depicts a class diagram created in the Unified Modeling Language (UML) to illustrate the graph representation of the information collected from the DEXPI XML file (C1) and process component and pipeline lists (B3). It is demonstrated that each item in the graph representation requires a unique set of attributes in order to identify and label the item. Attributes, such as type, location, and rotation, can assist the software in modeling the system more accurately.

Utilizing the algorithm provided in [9], a graph model (C2) based on the DEXPI XML file (C1) was created. The original algorithm was improved, and it can now filter out more information as well as detect additional attributes, such as XY position and rotation of the components.

The output from step (B3) comprises the process components, their descriptions, and the pipeline connections between the components in text format. Using this information, node and edge lists can be generated in .CSV format. This graph model (B4) contains only the elements that are essential for a steady state simulation of the system.

There are two graph models available at this point that can be used for the generation of a common model. The roadmap can be created in a variety of ways depending on the application, the requirement, and the available data resources in each model. With respect to the current case study, and in accordance with our goal of developing a model for steady state simulation, the graph model generated using the component and pipeline lists was considered as a primary model, while the graph model generated from DEXPI was considered as a supporting model for extracting the necessary information. As a result, the algorithms for model combination were created in such a way that they first check the steady state graph

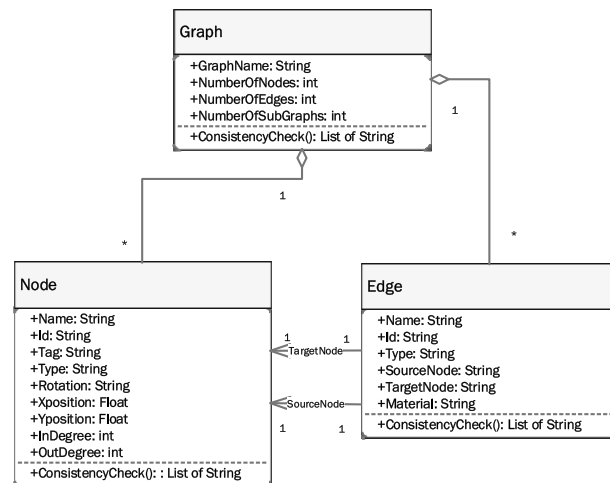


FIGURE 5. UML Class diagram of the graph representation of the information extracted from the Proteus XML file.

and then the DEXPI graph. Figure 6 depicts the algorithm for extracting the “type” attribute for a component from the steady state graph. As illustrated, the search algorithm checks the steady state graph model first to determine the type of equipment, and if that info is not found, it checks the DEXPI graph model. If a node type is not present in either of the models, we expect to be able to determine the type by analyzing the letters in the tag.

As previously stated, the DEXPI graph model can be regarded as a supporting model that provides the attributes required by the application. In some situations, this role is extremely vital because there are no other available resources for the required information. For example, the X and Y coordinates (XY), which indicate the object’s horizontal and vertical position in the flowsheet, can only be achieved by exploiting the DEXPI graph model. This information can be utilized to automate the locating of items in the simulation software as well as the visualization of the digital twin automatically. Sometimes, due to a missing component caused by the *DEXPI_XML_Generator*’s inability to recognize the customized objects in the P&ID, it is not possible to obtain direct access to XY coordination. As seen in Figure 7, the algorithm will look for a tag that is similar to the name of the component in order to determine its location. This approach is efficient because the *DEXPI_XML_Generator* can efficiently recognize all of the text in the flowsheet, but it cannot detect all of the objects within the flowsheet.

Finally, the system’s common graph model, which includes all required attributes for steady state simulation, is complete. This graph is made up of two parts. The two .CSV files include a node and attribute list, as well as an edge list.

F. SIMULATION SOFTWARE SPECIFIC MAPPING RULES (D1)

In the proposed approach, the second manual action performed by the human expert is the formulation of the

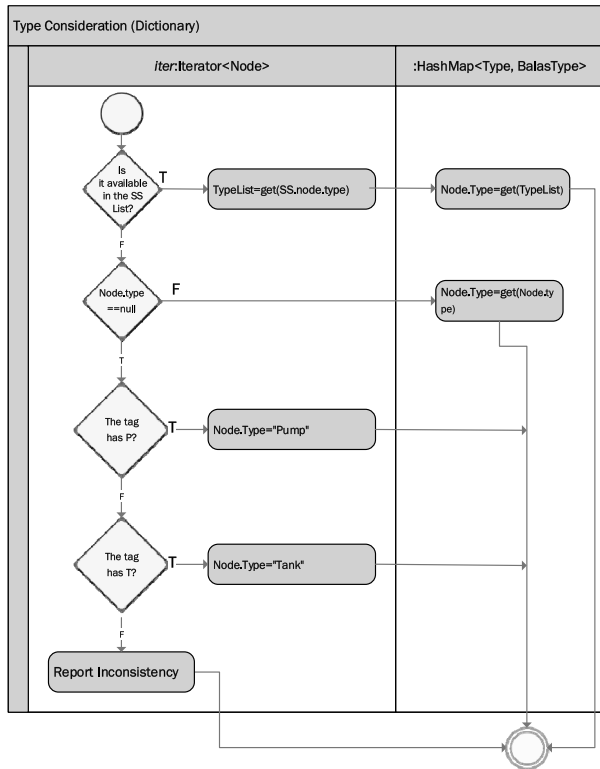


FIGURE 6. UML activity diagram representation of an algorithm that considers the type of a process component based on information that can be extracted from different sources.

simulation software specific mapping rules. These rules capture the modeling knowhow, i.e. the tacit knowledge, of the human modeler. They define how each process component selected from the P&ID using the Component_Selector will be presented in the selected simulation software. The rule defines what kind of simulation symbol(s) will be used in the simulation software to describe a process component, e.g., a tank or a pump. Even though the building of a simulation model using different commercial simulation software follows the same path (dragging and dropping symbols from libraries and drawing streams connecting them), the appearances and the calculation principles of the symbols may differ from software to software. Thus, the rules are always valid only for the selected simulation software. Also, since the rules are written by humans, there may be as many ways of writing the rule as there are writers.

G. REQUIRED INPUT FOR FLOWSHEET POPULATION (D2)

During the scale-up work of the methodology, the mapping rule library has expanded resulting in a notable increase in the number of process components that can be mapped with Balas simulation symbols as illustrated in the inherited class components in Figure 8.

New rules for mapping the new and more complicated process components with corresponding Balas simulation symbols, such as *Deculator*, *Screen*, *Mixer*, *Manifold*, *Heat exchanger*, *Dewatering element*, and *Headbox*, have been

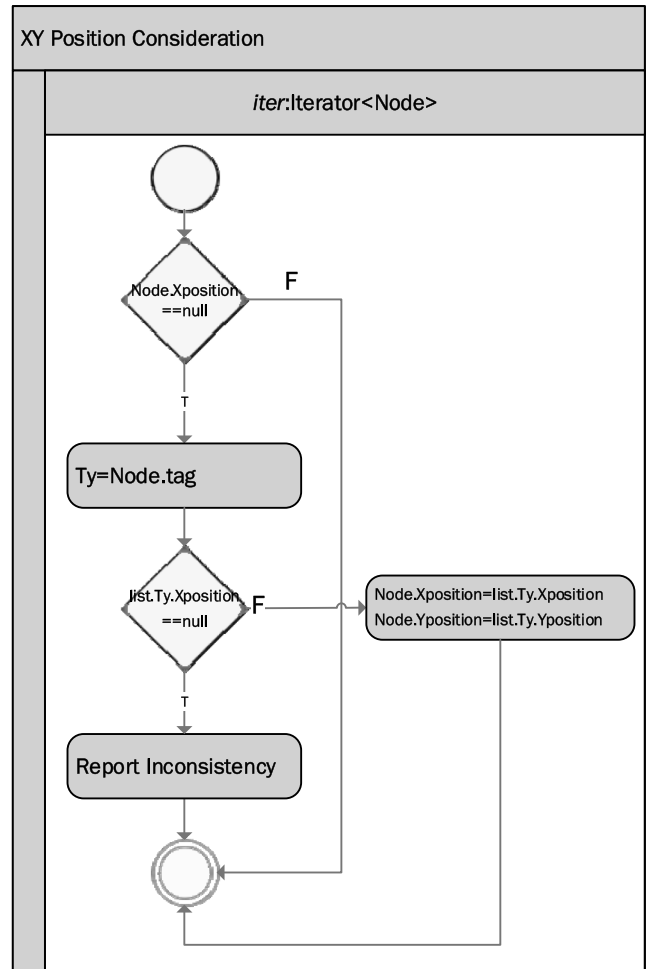


FIGURE 7. UML activity diagram representation of an algorithm that extracts information related to a position of the process component in a P&ID file. This info can be used for locating elements in the final model.

added in comparison to previous efforts [9]. These new rules are more challenging, not only because of the increased number of inputs and outputs, but also because several of the new process components have new “stream” parameters that make the port assignment more difficult. In the case of the *Heat exchanger* symbol, for example, the hot and cold inputs and outputs must be assigned to specified ports, whereas in the case of tanks port numbers can be assigned randomly. This solution is depicted in the UML activity diagram shown in Figure 9.

It is not possible to connect certain two simulation symbols directly in the Balas simulator. In some circumstances, a bridge component is required to make the connection possible. For example, an outlet stream from a tank (*MDTank#4*) cannot be connected directly as an inlet stream to another tank. Instead, they must be connected via a *Pump* symbol. A *Manifold* should be considered after and before a *Pump* with multiple inputs or outputs. In order to simulate non-storing connections and fitting elements having more than two input and output, such as *Mix2Tee* and *Manifold*, *Valves* must be added. The mapping rules and the Graph_Generator

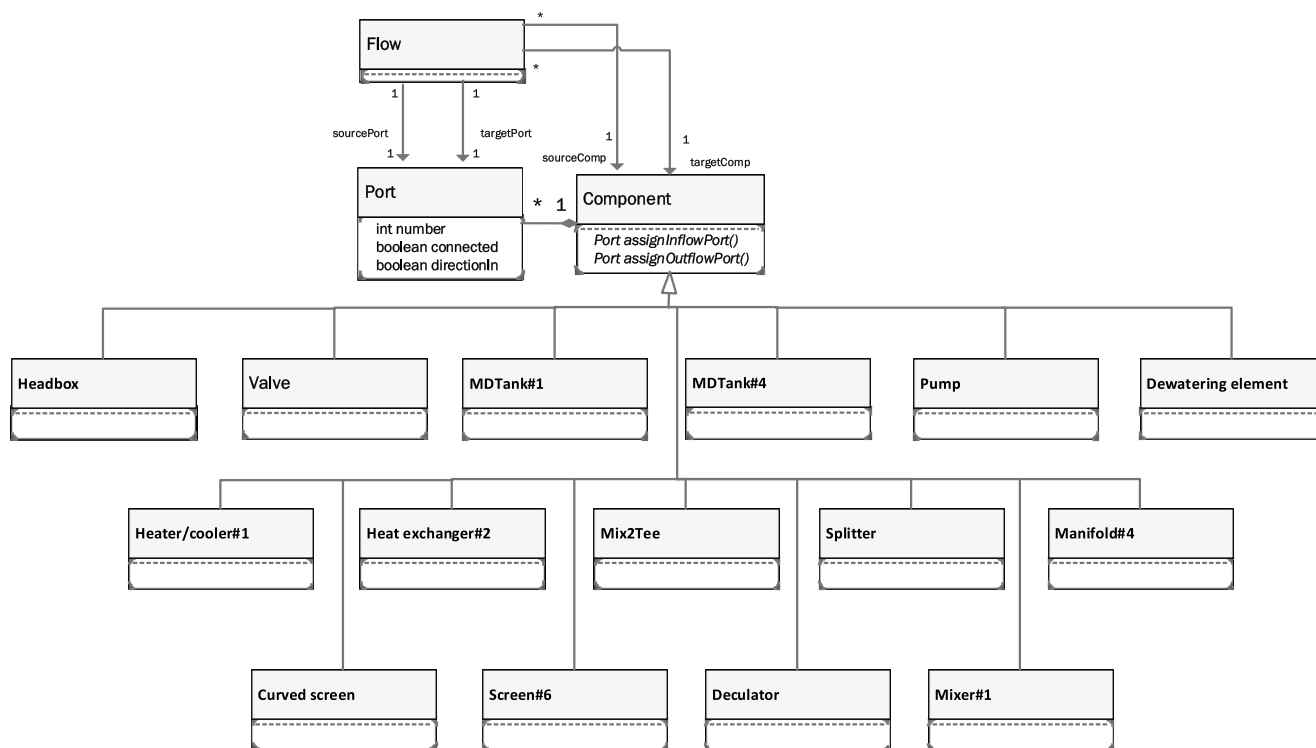


FIGURE 8. UML Class diagram of the elements in the graph representation of the system for Balas simulator.

have taken into account all of the previously listed items for adding bridge components. Figure 10 depicts a UML class diagram that shows how new nodes can be created with identical attributes. Also, the classes in this UML have attributes for port numbers which make them ready for port assignment.

The bridge components lack position or rotation attributes. The majority of process components in any drawing or HMI system have a “neutral” rotational orientation (zero degree). Also, manual modifications are required at various stages in the methodology’s subsequent steps. So, for all bridge components, this work considers a neutral zero-degree rotation, and a possibility, if needed, for the human modifier to adjust the rotations in the simulation software to make the visualization more structured.

Regardless of the rotation, achieving proper XY coordination may be time-consuming and necessitates the aid of someone who is familiar with the system and can understand the process philosophy. Thus, automatic locating of bridge and self-named components will accelerate the process of creating digital twins. Some of the known nodes (mainly derived straight from DEXPI) have specific coordination in the Cartesian coordinate plane, as seen in Figure 11. The bridge, as well as the self-named components with unknown coordination must be located in the same XY plane. “APPENDIX A” discusses the most common solutions for achieving XY coordinates for the unknown nodes.

H. FLOWSHEET POPULATOR (D3)

Flowsheet population is based on major unit operations and streams between them. The input information for the *Flowsheet_Populator* (D3) is provided as a JSON file including the graph nodes (process components) and edges (pipeline connections). The *Flowsheet_Populator* reads the input file and generates unit operation symbols and their connections on a given target flowsheet. In our approach the flowsheet is visualized in MS Visio.

An implemented population algorithm is quite simple, because the input information is already advanced processed. Main phases of the algorithm are:

- loop over all nodes
 - select a type of unit operation according to given node type
 - create unit operation symbol on a default location
 - create and set a valid unit operation, its identifier and name based on given node identifier
 - update maximum and minimum encountered location coordinates
 - set and use the default calculation module for the unit
- loop over all edges
 - create a stream
 - select input and output ports of unit operations according to predefined port assignments
 - glue the stream to the output and input ports (automatic routing is used)

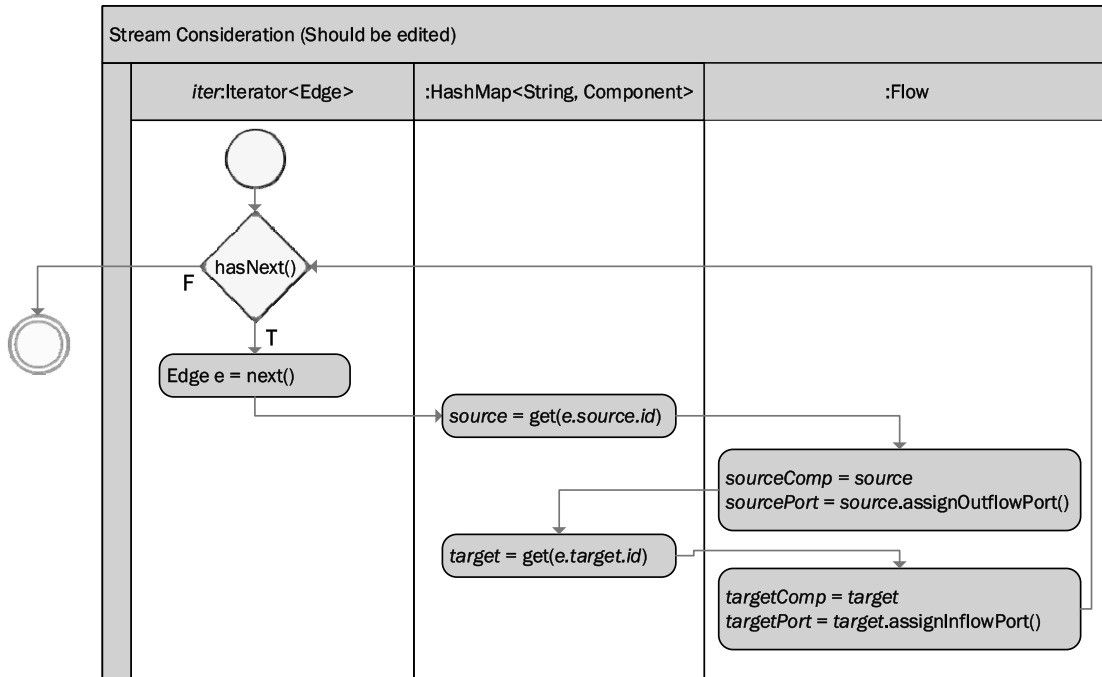


FIGURE 9. UML activity diagram representation of an algorithm that creates the edges and connects them to the correct ports according to the mapping rules and stream order.

- loop over all unit operations
 - mirror, move and scale location coordinates of unit operation symbol depending on encountered minimum and maximum coordinates

Software architecture of the Flowsheet_Populator is shown in the Figure 12.

The implementation applies mediator and factory design patterns, and is also based on parser technologies. Balas simulation software (client) creates a mediator (JSONMediatorFactory) and calls its import method to populate a flowsheet. The mediator (or a factory) creates and initializes its components. A scanner (JSONScanner) scans an input file and returns a stream of tokens by the aid of a token manager (TokenMgr). The mediator sends tokens to a parser (JSONParser) for input validation. Errors are reported by a logger (Logger). During validation, the parser sends events via the mediator to a generator (JSONGenerator) which finally populates a flowsheet (D4).

I. GENERATION OF STEADY STATE SIMULATION MODEL FLOWSHEET (D4)

Balas simulation models are created and maintained through an intuitive user interface, namely MS Visio. The models are built up by dragging and dropping unit operation symbols from libraries, drawing streams connecting the symbols, and entering input data using dialog windows. Since the human expert performs this normally manually, it is quite time consuming. Our approach replaces the manual phase of the flowsheet generation by generating the flowsheet semi-automatically using the Flowsheet_Populator developed in

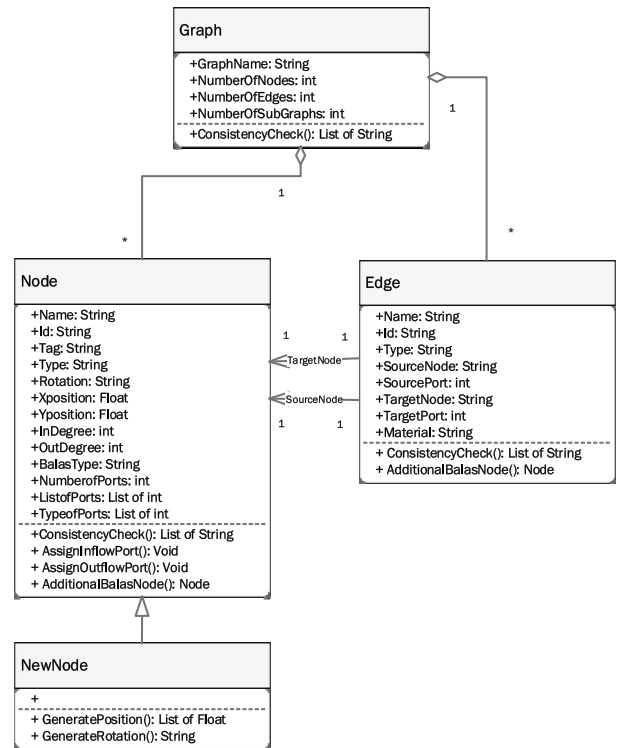


FIGURE 10. UML Class diagram of the graph representation after implementing the mapping rules.

D3. The Flowsheet_Populator is added as an Add-On to MS Visio, and the human expert calls the Add-On to generate the flowsheet. The generated flowsheet is “printed” on the

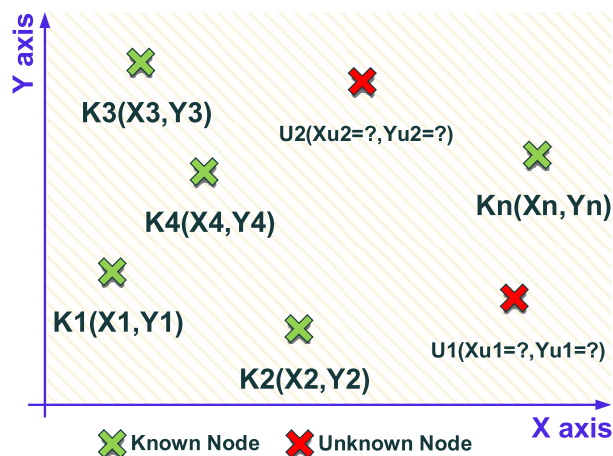


FIGURE 11. Location of the known and unknown nodes in the P&ID flowsheet.

left from target sheet and can be later moved to correct place in order to enable multiple input files population on a same sheet.

J. FROM DIGITAL FLOWSHEET THROUGH DIGITAL MODEL TO DIGITAL TWIN (D5–D8)

The steady state simulation model (D6) is achieved by manual initialization and parametrization (D5) of the flowsheet generated with the Flowsheet_Populator (D4). The human expert can make the automatically populated flowsheet more visually attractive by replacing the unit operation symbols and rerouting the streams. Manual initialization of the model means selecting the calculation modules for the unit operation symbols and the chemical components present in the process (i.e. water, fiber, chemicals). Each symbol may have optional calculation modules, which the human expert selects manually in the simulation software from a drop-down list. After initialization comes the parameterization of the calculation modules. Each calculation module determines a set of input values, i.e. parameters that are needed to parameterize the module. These include, e.g., pressure, temperature, flow, consistency, retention or removal degree. According to Kritzinger’s definition [3], the digital model (D6) converts to a digital twin (D8) not until there is a two-way interconnection (D7) between the physical and digital system. The two-way communication between the physical and digital system can be automated through a manufacturing execution system (MES) or process automation system.

IV. CASE STUDY

The above proposed methodology for the semi-automatic generation of a steady state digital twin was demonstrated with a case study. The case study was VTT’s (VTT Technical Research Centre of Finland Ltd.) SUORA paper and board making research facility (Figure 13). It offers cost efficient prototyping of ideas, fast experimenting, and development of new process solutions. SUORA has about 600 measurement points that are connected to the process’ data control

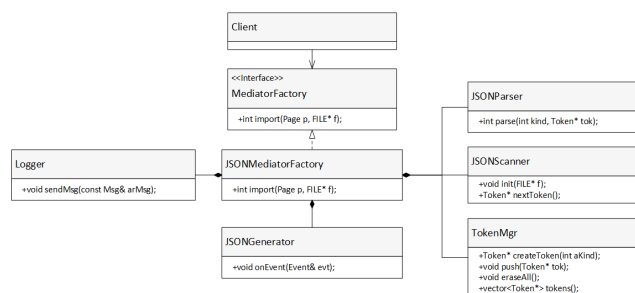


FIGURE 12. Flowsheet_Populator software architecture.

system (DCS). It is depicted with three P&IDs. However, our methodology is demonstrated using only one of the P&IDs. The application of the methodology for several P&IDs was demonstrated during this work, but not reported here.

V. RESULTS AND DISCUSSION

A. P&ID CONVERTOR (B1) AND COMPONENT_SELECTOR (B2)

The case study’s P&ID was in AutoCAD file format. It was converted to a more open format by printing the AutoCAD file into a PDF file. The Component_Selector program recognized and extracted text fields in the PDF file by using the open-source tool Tesseract 5.0. The results were not perfect since all text snippets were not recognized, and some diagram elements were interpreted as letters. However, the results were sufficient for our purposes. The human expert uses the Component_Selector to select those process components that are wanted to be included into the steady state simulation model by clicking the name of the component inside the extracted rectangles. Selected components are shown with red rectangles. The user can also remove any component selection if a mis-selection was made. After selecting all the needed process components, the user pairs any two of the selected components for creating the connective pipelines. During the pair selection, the user clicks first the extracted text field of the source component of the pipeline and then the extracted text field of the target component of the pipeline. The source components are shown with green rectangles (inside the red rectangles), whereas the target components are shown with blue rectangles (inside the red rectangles). Besides the colored rectangles, the pipeline connections from sources to targets are shown with blue lines. The Figure 14 shows how two selected components look like after one pairing has been created. The case study’s P&IDs contained also components with no tag names making it impossible for the Component_Selector program to recognize and extract the text fields. In this case, the user was able to create and name an own text field, i.e. create self-named process components.

B. GENERATION OF COMPONENT AND PIPELINE LISTS FOR A STEADY STATE SIMULATION MODEL (B3)

The process components selected with the Component_Selector were written to the text file “items.txt” that contained the IDs for the selected components. The text file

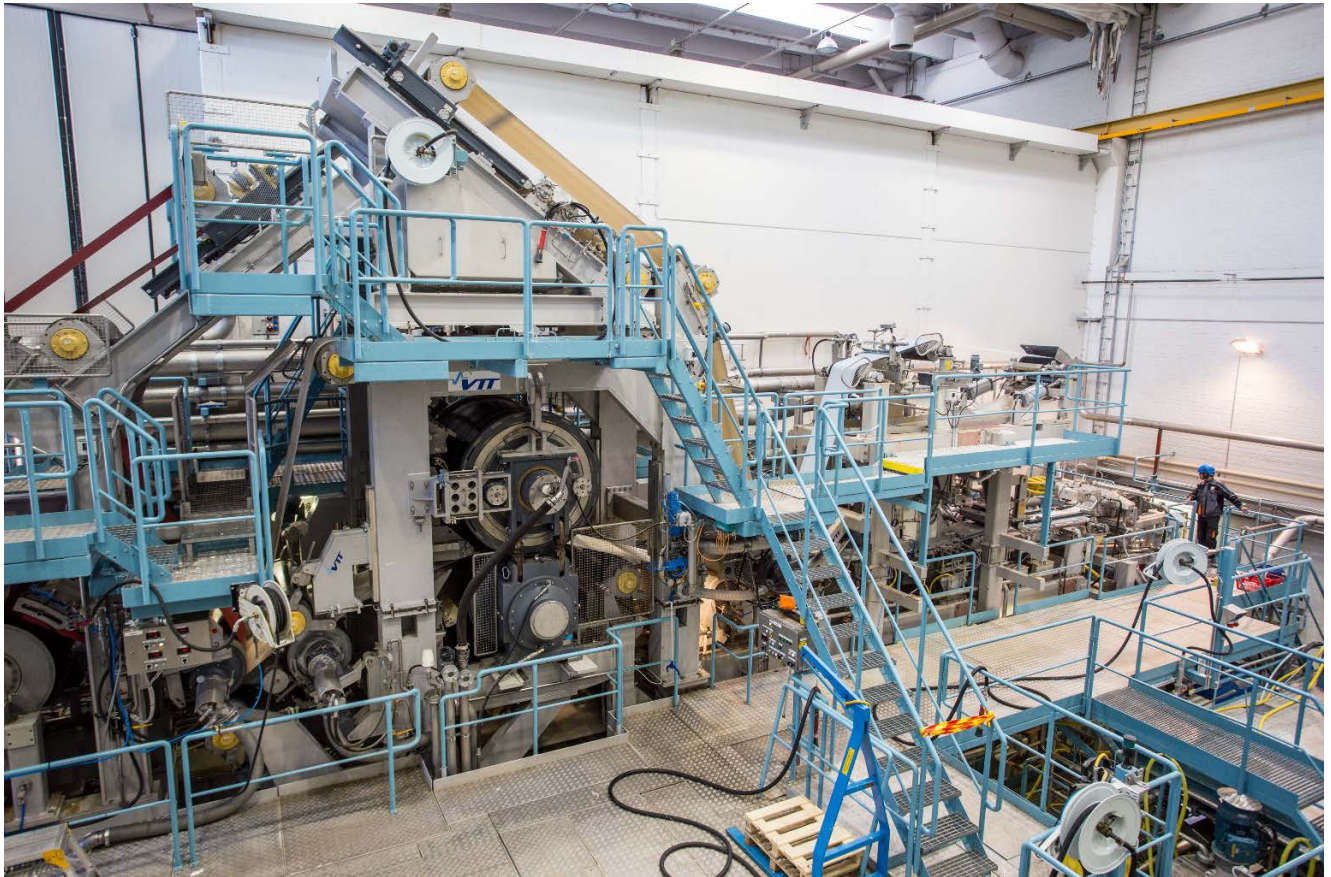


FIGURE 13. SUORA paper and board research facility.

TABLE 1. Part of the extracted process component list generated using component_selector.

PROCESS COMPONENT	Type
822-T-003	Deculator
822-T-001	Tank
822-P-010	Pump
822-E-012	Mixer
Headbox	Headbox

(NOTE1: Self-named equipment: Headbox)

also contained human expert's manual descriptions for component types in cases where the *DEXPI_XML_Generator* wasn't able to recognize the symbol from the P&ID and map it to corresponding DEXPI element. Correspondingly, the selected pipelines were written to the text file "pairs.txt" that contained the ID of the source component followed by the ID of the target component. Examples of these files are presented in Table 1 and Table 2. They were then used as input files in step B4, i.e., graph processing.

TABLE 2. Part of the extracted pipeline list generated using component_selector.

SOURCE COMPONENT	TARGET COMPONENT
822-T-001	822-P-005
822-E-012	822-P-008
822-P-010	822-E-012
822-T-001	822-T-002
822-T-003	822-E-012

C. DEXPI XML GENERATOR (C1)

With *DEXPI_XML_Generator*, most of the process component symbols, pipeline connections and attributes could be recognized and mapped to corresponding elements in DEXPI. The recognition percentage for elements including pipeline components, automation components and equipment for the case study's three P&IDs varied from 88% to 94%. Examples of recognition results can be seen in Figure 15. A short summary of cases that were difficult to recognize is given below. As for textual content, the labels for equipment located



FIGURE 14. The Component_Selector application assists human experts in recognizing and selecting the desired components in the P&ID document¹.

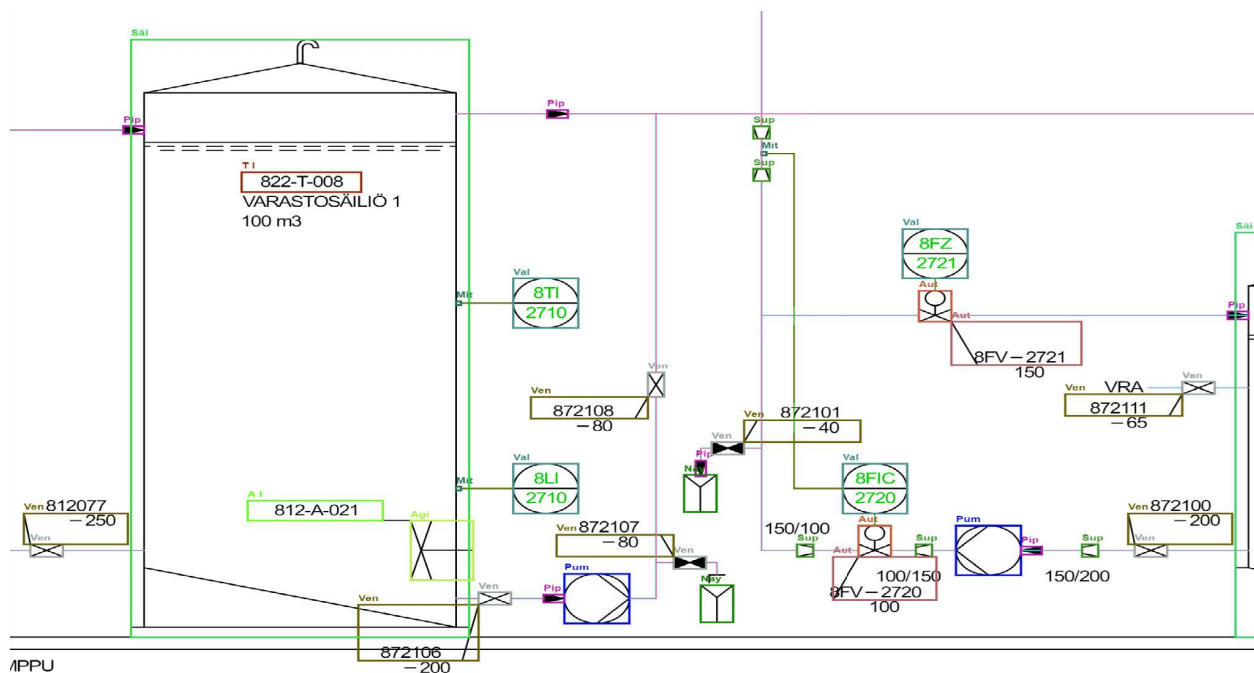


FIGURE 15. Example of recognition results of the DEXPI_XML_Generator. A bounding box has been drawn around the graphics of the recognized symbols and label lines.

inside box shapes could be reliably recognized. The labels for valves, on the other hand, did not have a consistent shape, as the length and angle of the label line depended on the placement of the label in regard to the associated valve. Some of these labels were left unrecognized, but most could be found.

Some of the more complex equipment symbols were also difficult to recognize. For example, the case study’s P&IDs included large equipment symbols that in some cases included other symbols inside them (Figure 16 (a)). Hoses (Figure 16 (b)) and pipeline arrows (Figure 16 (a)) could have rotations other than the supported 90-degree rotations and horizontal flips, and in these cases, they were not recognized. Either a new recognition pattern should be created for the unrecognized rotations, or support for recognizing symbols with arbitrary rotations should be implemented.

¹To protect the confidentiality of the case study, the figures in this paper have been carefully modified to remove any confidential information, such as pump information, special piping fittings and valve and pipe dimensions.

Typically, a single diagram will use the same style to represent all line crossings. DEXPI_XML_Generator can be configured to support one of the three following options: gaps, overlapping lines or a symbol, such as a small arc located at the point where the lines cross. In the case study’s P&IDs a small gap was used most of the time, but in few locations, there was some inconsistency, which meant that some of the line crossings could not be handled correctly. There were cases where the gap was either missing (Figure 16 (c)) or the gap size was significantly larger than usual (Figure 16 (d)).

Finally, the DEXPI files (example in Figure 17) were created and used in the step C2. Depending on the complexity of the P&ID, DEXPI files may include millions of lines.

D. GRAPH PROCESSING (B4, C2)

Figure 17 shows a DEXPI file (C1) in XML markup language that was used as an input file for the Graph_Generator (C2). The Graph_Generator extracted from this file general information about the process components, such as the type,

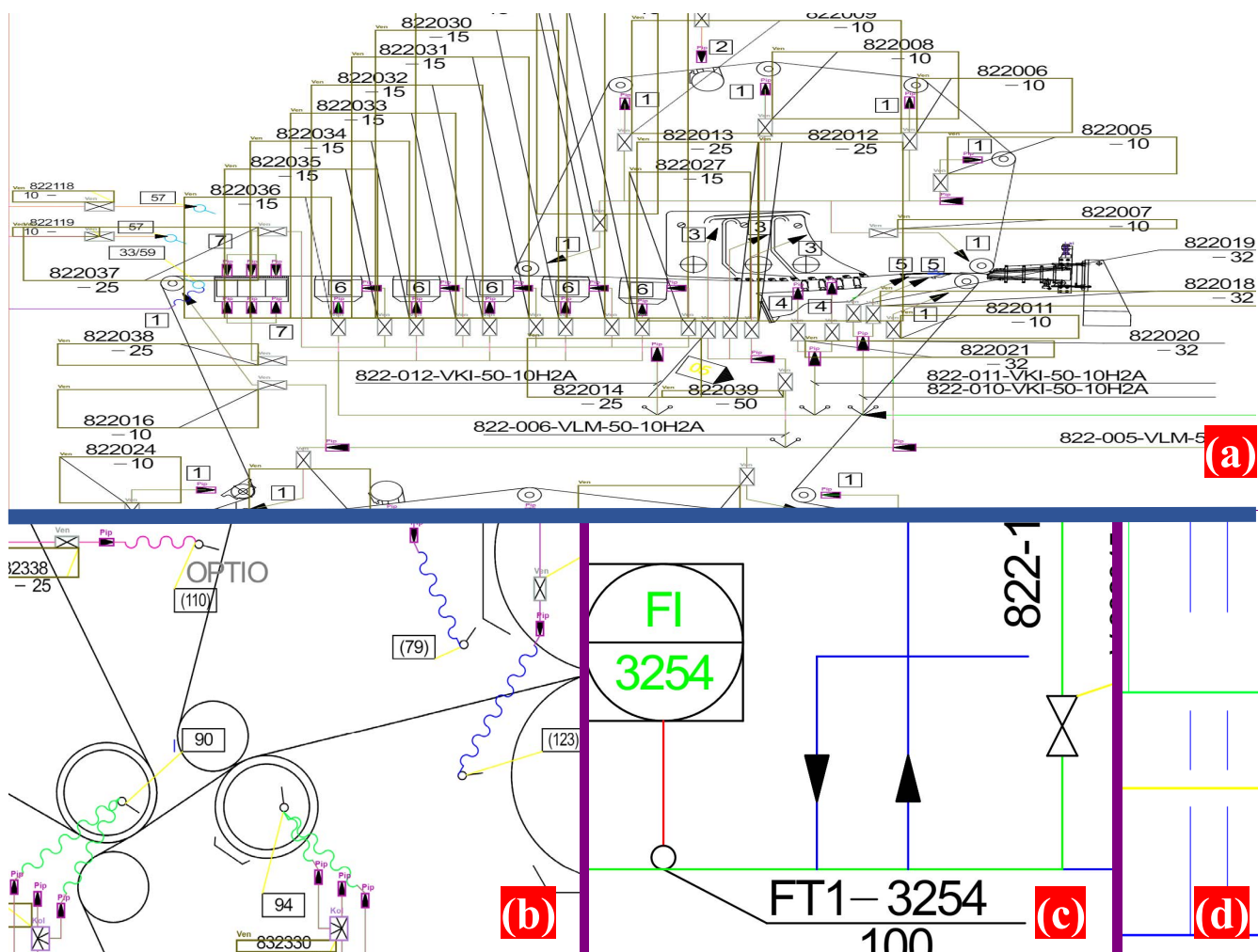


FIGURE 16. Pattern recognition is not straight forward and can be affected by misrecognition or failure: (a) Example of a complex equipment shape and unrecognized pipeline arrows. (b) Example of unrecognized hose elements. (c) Example of a case where the gap is missing from the point where the blue lines cross each other. (d) Example of a case where the gap size is inconsistent.

position, and rotation of the component in the engineering document, as well as the connections between distinct components. Table 3 and Table 4 provide examples of the raw data derived from the DEXPI file of the case study using the *Graph_Generator*.

The *Graph_Generator* (B4) was also applied to construct another parallel graph model. In this case, the process component and pipeline lists created with the *Component_Selector* (B3) were used as input files. Because this graph model provided all of the essential node and edge data for a steady state simulation model, it was used as the base model while the C2 graph model provided required attributes.

E. SIMULATION SOFTWARE SPECIFIC MAPPING RULES (D1)

The mapping rules define how each graph node derived from selected process components of P&ID (e.g. tank, heat exchanger, screen) will be presented in the selected simulation software, in our approach, Balas steady state simulation

TABLE 3. Part of the extracted node list from the Proteus DEXPI XML file. It includes attributes, like XY coordination, and class type. The complete list has hundreds of components.

Node	Tag	Class	X	Y
Name		(In Finnish)	position	position
N2	822-E-012	Vedenerotin	7160.28	1187.577
N8	833-E-008	Äänenvaimennin	4420.339	1168.894
N12	822-T-018	Vedenerotin	8217.04	1130.342
N73	833-P-009	Pumppu	1027.625	290.2764

software. The rules act like an interpreter between the original P&ID and the simulator. They name the Balas unit operation symbol and define to which ports of the symbol the inflows and outflows are connected. Table 7 in Appendix B shows examples of Balas specific mapping rules. These mapping rules are used as input in step D2.


```

278908 <Min X="3730.532470703125" Y="28.266830444335938" />
278909 <Max X="3730.532470703125" Y="73.63347625732422" />
278910 </Extent>
278911 <Coordinate X="3730.532470703125" Y="39.608497619628906" />
278912 <Coordinate X="3730.532470703125" Y="28.266830444335938" />
278913 </PolyLine>
278914 <PolyLine NumPoints="2">
278915 <Presentation LineType="Solid" LineWeight="0.7087483406066895"
278916 <Extent>
278917 <Min X="3730.532470703125" Y="28.266830444335938" />
278918 <Max X="3730.532470703125" Y="73.63347625732422" />
278919 </Extent>
278920 <Coordinate X="3730.532470703125" Y="28.266830444335938" />
278921 <Coordinate X="3730.532470703125" Y="39.608497619628906" />
278922 </PolyLine>
278923 <PolyLine NumPoints="2">
278924 <Presentation LineType="Solid" LineWeight="0.7087483406066895"
278925 <Extent>
278926 <Min X="3730.532470703125" Y="28.266830444335938" />
278927 <Max X="3730.532470703125" Y="73.63347625732422" />
278928 </Extent>
278929 <Coordinate X="3730.532470703125" Y="50.950157165527344" />
278930 <Coordinate X="3730.532470703125" Y="39.608497619628906" />
278931 </PolyLine>
278932 <PolyLine NumPoints="2">
278933 <Presentation LineType="Solid" LineWeight="0.7087483406066895"
278934 <Extent>
278935 <Min X="3730.532470703125" Y="28.266830444335938" />
278936 <Max X="3730.532470703125" Y="73.63347625732422" />
278937 </Extent>
278938 <Coordinate X="3730.532470703125" Y="39.608497619628906" />
278939 <Coordinate X="3730.532470703125" Y="50.950157165527344" />
278940 </PolyLine>
278941 <PolyLine NumPoints="2">
278942 <Presentation LineType="Solid" LineWeight="0.7087483406066895"
278943 <Extent>
278944 <Min X="3730.532470703125" Y="28.266830444335938" />
278945 <Max X="3730.532470703125" Y="73.63347625732422" />
278946 </Extent>
278947 <Coordinate X="3730.532470703125" Y="62.29181671142578" />
278948 <Coordinate X="3730.532470703125" Y="50.950157165527344" />
    
```

FIGURE 17. An example of the DEXPI file that was produced with the DEXPI_XML_Generator.

TABLE 4. Part of the extracted edge list from the proteus XML file.

Edge Name	Source Node	Target Node
E7	N7	N8
E8	N9	N7
E9	N5	N9
E11	N8	N12
E14	N8	N14

F. REQUIRED INPUT FOR FLOWSHEET POPULATION (D2)

Based on the mapping rules formulated in step D1, the final graph model (D2), which was created by integrating the two graph models from steps B4 and C2, was adjusted to provide the required input data for simulation software flowsheet population (D4). It was necessary to assign the port numbers of the unit operation symbols and establish some additional bridge components like manifolds or valves in order to make the graph model compatible with the Balas simulation software. The output of this step included a list of required nodes for the Balas steady state simulation model (Table 5) and assigned port numbers for the edges (Table 6). Figure 18 depicts a visualization of the produced model (given in Table 5 and Table 6).

G. FLOWSHEET POPULATOR (D3)

The Flowsheet_Populator seemed to be a quite robust tool because it is based on parser technologies and can thus recover from error situations. The populator was tested during the case study development and no performance issues were reported.

TABLE 5. Part of the generated node list for the steady state simulation model. It includes attributes like XY coordination, rotation, and corresponding balas symbols.

Node Name	X Position	Y Position	Rotation	Balas Symbol
822-T-003	8490.806	1314.924	U	Deculator
822-E-012	8305.183	232.5682	U	Mixer#1
812-P-011	3793.924	227.5421	D	Pump
812-E-007	568.3316	1482.213	U	Heat exchanger#2
Headbox	6000	350	U	Headbox

TABLE 6. Part of the generated edge list for the steady state simulation model. It shows the edges between different nodes in the model. Ports are assigned based on the type of the node and Balas mapping rules.

Source	Source Port	Target	Target Port
812-T-006	5	812-P-011	1
822-E-012	1	822-P-008	1
812-P-011	2	Manifold102	3
Headbox	5	822-T-002	6
In102	1	812-E-007	1
Valve103	2	812-E-007	3
822-P-010	2	822-E-012	2

H. GENERATION OF STEADY STATE SIMULATION MODEL FLOWSHEET (D4)

Figure 19 shows parts of the populated steady state simulation model flowsheet for the case study. The upper flowsheet depicts the original flowsheet created with the Flowsheet_Populator and the lower flowsheet after some manual adjustments were made.

I. FROM DIGITAL FLOWSHEET THROUGH DIGITAL MODEL TO DIGITAL TWIN (D5–D8)

This paper doesn't report the whole presented methodology. It ends after the flowsheet of the simulation model has been created in step D4. The manual initialization and parametrization (D5) of the flowsheet are common manual actions performed by the modeler, and thus not reported here. Also, the running of the simulation model for different scenarios (D6) was not the focus of this study, and thus not presented here. Conducting the final step in the methodology, i.e. the connection of the physical process to the digital model, was hindered due to restricted rights to connect the process automation system of the case study process. Thus, it is not reported in this paper.

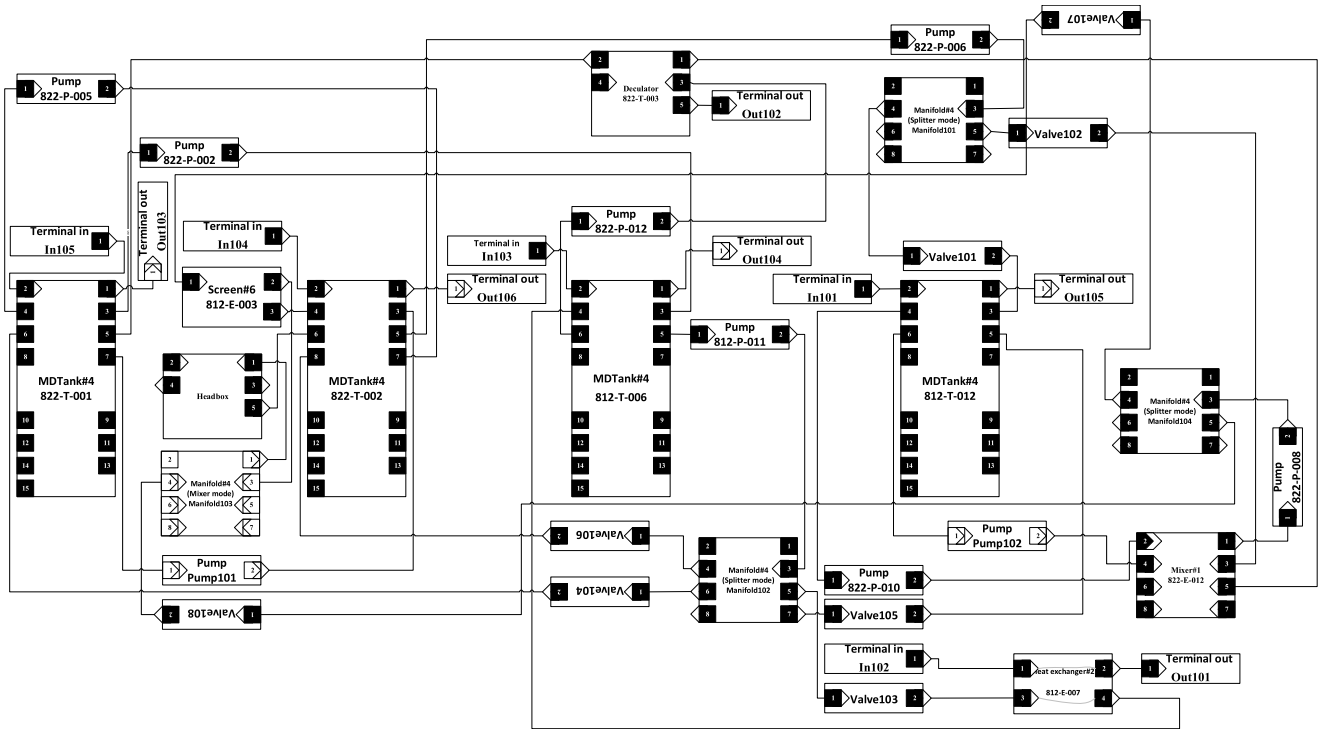


FIGURE 18. Manually visualizing and validating the steady state model after applying the Balas mapping rules (D2).

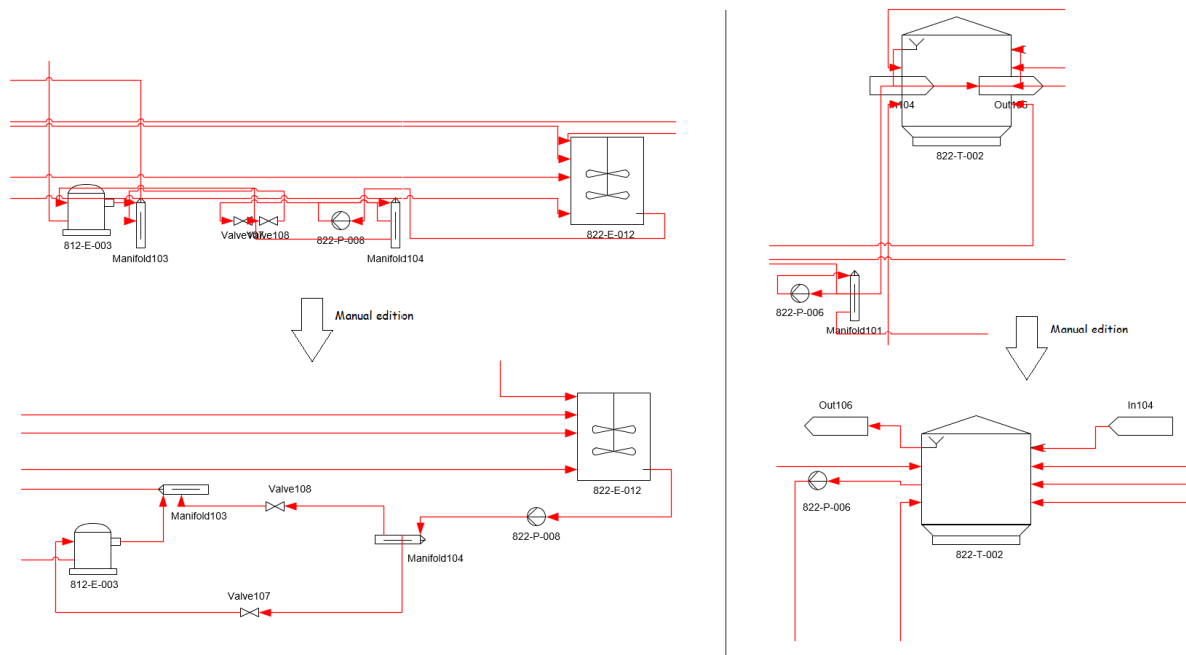


FIGURE 19. Before (up) and after (down) manual adjustment of the steady state simulation model flowsheet in the Balas software.

J. STUDY LIMITATIONS AND ALTERNATIVES

The research results given in this paper should be considered in the context of their limitations. In this section, the most significant constraints, alternatives, and solutions will be addressed in further detail.

- **Selecting of components from the P&ID is manual:** The Component_Selector tool has a simple user interface (UI), and many of its features were developed quickly as the demand occurred. Therefore, it recalls an expert to use it. This expert must also do lots of manual

work. If the UI of the tool was developed from scratch, it could be made accessible for less advanced users. Currently there is only one feature, which provides the possibility for “mass editing,” i.e., selecting a number of process components automatically based on their names. This “mass editing” feature should be expanded and generalized (and more of similar features should be invented), so tedious manual work could be avoided as much as possible.

- **Mapping rules are simulator specific and highly case sensitive:** The mapping rules are only valid for the simulator they are identified. Every process simulator has unique unit operation symbol libraries. If a new process component is found in the P&ID that does not appear in the mapping rule library, the software in step B4 will not be able to discover a suitable component for simulation software and assign ports to it. To overcome this constraint, a general rule for unrecognized components must be considered. They can be described with a general simulation symbol, i.e. Undefined sub-process, that the human expert manually adjusts.
- **Object detection may be inefficient under certain circumstances:** As described in Sect V.C, object detection might fail because of complex symbols, arbitrary rotations or inconsistencies and errors in the source data. Improving the object detection and the quality of the source data would increase the number of correctly detected symbols. In the meanwhile, missing information can be handled by manual correcting the DEXPI file or by utilizing the graphical and textual information of unrecognized symbols saved under the Drawing element of the DEXPI file.
- **Tools are domain specific:** Although this research took place in the paper industry, and most of the tools were built using components accessible in that sector, the main methodology is generic and may be applied to other sectors of the process industry. By adding new components to the developed tools and software’s libraries, it would be straightforward to upgrade them for new domains.
- **DEXPI_XML_Generatoris working for vectorized P&IDs:** For *DEXPI_XML_Generator*, support for raster images and DWG as source formats in addition to vectorized PDF are currently under development. DWGs can already be utilized if they are first exported into vectorized PDFs but using DWGs directly as a source format would allow the recognition to access more information during the recognition, such as block structures and layers. The raster image support would make it possible to translate scanned documents in addition to vectorized PDFs.

VI. CONCLUSION AND FURTHER WORK

This paper describes an approach for extracting semi-automatically required process information from engineering documents, such as Piping and Instrumentation Diagram

(P&ID), for generating a steady state simulation model and creating a digital twin for a paper process system. This paper proposed a generalized methodology that has resulted in the development of a number of tools and software for the semi-automated generation of digital twins. They were developed to pave the way for a more automated solution for creating a model based digital twin; *DEXPI_XML_Generator* for creating a DEXPI model, *Component_Selector* for assisting a human expert in selecting desired process components and pipelines from the P&ID, *Graph_Generator* for creating an intermediate graph model, and *Flowsheet_Populator* for generating flowsheets required for a simulation software. The value of this work is that a complex workflow involving a long software chain has been implemented and demonstrated. Many of the software tools in the chain are rudimentary, ad hoc developed, and have a simple user interface that can be used to gradually ease the transition to digital twins. The effectiveness of the proposed methodology was shown by the reported step-by-step outcomes on a pilot paper machine case study. The results showed that an initial investment in the development of automated solutions might lead to the creation of digital twins faster, more cost-effective, and less human-dependent.

The current study, like most others, has limitations that are listed in Section V.K, such as domain limitation and robustness against uncertainties and incorrect inputs. There are also some plans in place for the work’s future expansion. In the future, several forms of inputs, as well as simulation software suited to industry needs, may be investigated. The system’s domain must be expanded using different use cases, and its findings must be proven in real-world industrial applications. Also, interconnection to the plant’s DCS to transform the digital model to a digital twin could be done automatically. In addition, it would be feasible to do a real-time update of steady state model parameters based on recent measurements of the process.

APPENDIX A. POSITIONING UNKNOWN COMPONENTS

As mentioned in the section III.G, some nodes in a P&ID document have known XY coordination, while some nodes with unknown XY coordination, such as freshly generated components based on Balas rules or newly named components, must be positioned in the XY plane. An unknown node connected to two or more known nodes, two or more unknown nodes connected in series in the middle of known nodes, or special nodes connected to specific components are only some of the possibilities. In all circumstances, the suitable position for unknown nodes should be determined. This appendix will go over the most common situations and the best ways to deal with them.

A. AN UNKNOWN COMPONENT BETWEEN TWO KNOWN COMPONENTS

Tank symbols can’t be linked directly together via direct pipeline according to Balas mapping rules. Instead, a pump symbol must be added in between. The required attributes

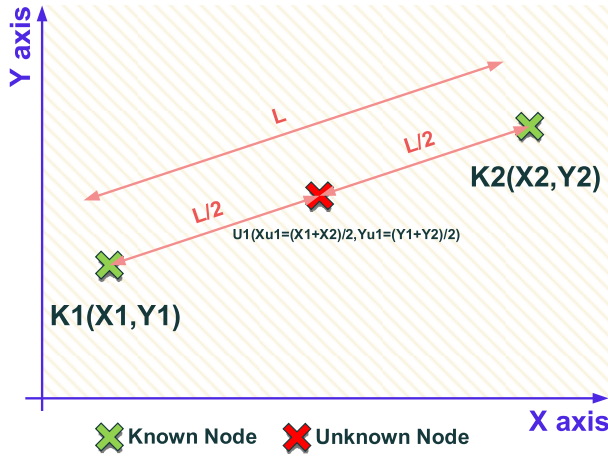


FIGURE 20. Positioning a new component between two known components.

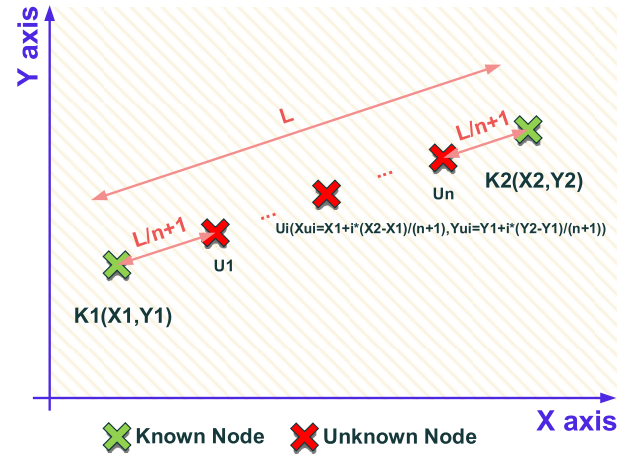


FIGURE 22. Positioning several new components between two known components.

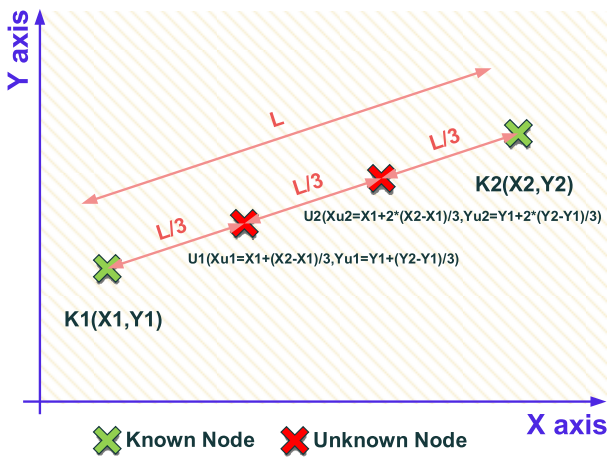


FIGURE 21. Positioning two new components between two known components.

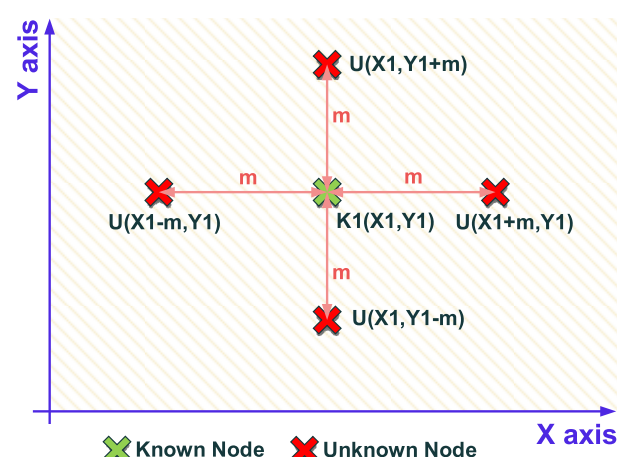


FIGURE 23. Using a fixed distance for a new component to be located next to a known Component.

for this pump, such as XY coordinates, should be created. Generally, to locate an unknown symbol between two known symbols, we may place it in the midpoint of the known symbols. If the coordinates of the first known symbol or point are $K_1(X_1, Y_1)$ and the coordinates of the second known point are $K_2(X_2, Y_2)$, then the coordinates of the connected unknown point $U_1(X_{U1}, Y_{U1})$ as shown in Figure 20 can be found using formula (1).

$$U_1(X_{U1}, Y_{U1}) = \left(\frac{X_1 + X_2}{2}, \frac{Y_1 + Y_2}{2} \right) \quad (1)$$

B. SEVERAL UNKNOWN COMPONENTS BETWEEN TWO KNOWN COMPONENTS

In a case where we need to insert two unknown symbols or points, $U_1(X_{U1}, Y_{U1})$ and $U_2(X_{U2}, Y_{U2})$, between two known points, $K_1(X_1, Y_1)$ and $K_2(X_2, Y_2)$, in a serial form, the coordinates of the unknown points as shown in Figure 21, can be found using formula (2).

$$U_1(X_{U1}, Y_{U1}) = \left(X_1 + 1 * \frac{X_2 - X_1}{3}, Y_1 + 1 * \frac{Y_2 - Y_1}{3} \right)$$

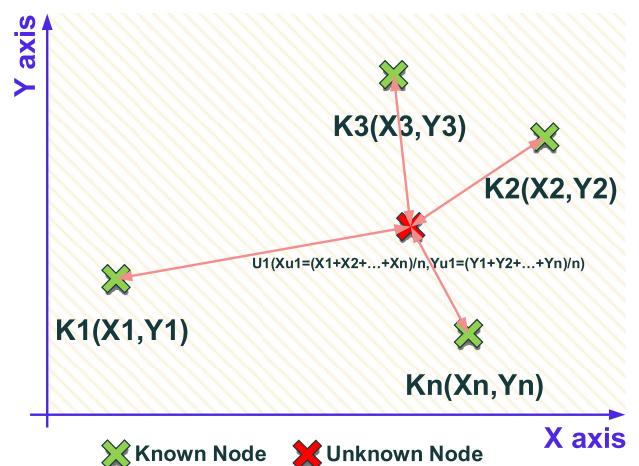
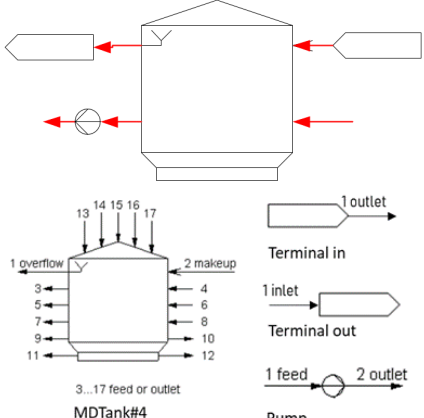
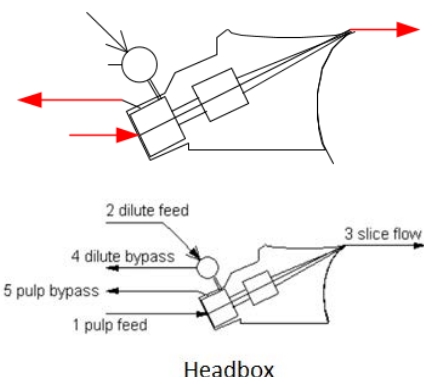
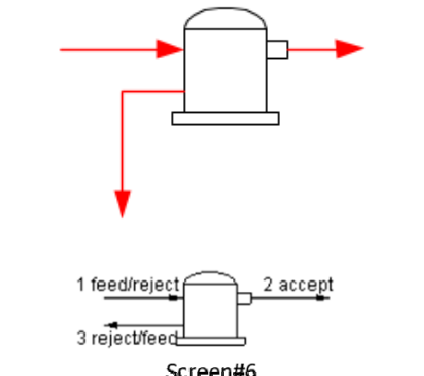


FIGURE 24. Positioning a new component connected to several known components.

$$U_2(X_{U2}, Y_{U2}) = \left(X_1 + 2 * \frac{X_2 - X_1}{3}, Y_1 + 2 * \frac{Y_2 - Y_1}{3} \right) \quad (2)$$

TABLE 7. Example of balas mapping rules.

Graph structure	Mapping to Balas steady state simulation software	
<p>A node of type tank, with several outgoing and incoming edges.</p>	<p>Replace all tank nodes with a symbol "MDTank#4". Add a stream from port #1 of symbol "Terminal in" to port #2 of symbol "MDTank#4". From port #1 of symbol "MDTank#4", add a stream to port #1 of symbol "Terminal out". Ports #3-#17 of symbol MDTank#4" can be used either for feed or outlet so that at least one stream enters port #3...#17 and one streams exits port #3...#17. For an exiting stream, add a stream from any free port #3...#12 of symbol "MDTank#4" to port#1 of symbol "Pump". Add a stream exiting the port#2 of symbol "Pump".</p>	
<p>a node of type headbox with one incoming edge and two outgoing edges</p>	<p>Replace a headbox node with a symbol "Headbox". Add a stream entering the port#1 of symbol "Headbox". Streams exiting the "Headbox" are numbered as stream #1 and stream #2. Add stream #1 exiting port#3 and stream #2 exiting port#5 in symbol "Headbox".</p>	
<p>a node type of machine screen</p>	<p>Replace a machine screen node with a symbol "Screen#6". Add a stream entering the port#1 of symbol "Screen#6". Streams exiting the "Screen#6" are numbered as stream#1 and stream#2. Add stream #1 exiting port#2 and stream #2 exiting port#3 in symbol "Screen#6".</p>	

This can be extended, to distribute n points between two known points, $K_1(X_1, Y_1)$ and $K_2(X_2, Y_2)$, so that the distance between every consecutive points is equal. the coordinates of the unknown point i, where $1 \leq i \leq n$ as shown in Figure 22, can be found using formula (3).

$$U_i(X_{Ui}, Y_{Ui}) = \left(X_1 + i * \frac{X_2 - X_1}{n + 1}, Y_1 + i * \frac{Y_2 - Y_1}{n + 1} \right) \tag{3}$$

C. USING A FIXED DISTANCE

According to Balas mapping rules, some symbols must always be considered next to other symbols, such as "Terminal In" and "Terminal Out" symbols which are mandatory for a tank symbol or in a case of series of dewatering symbols. In this situation, the secondary symbol can be placed to the primary symbol at a fixed distance. The secondary symbol can be placed horizontally or vertically before or after the primary symbol, depending on the

primary symbol's default format and rotation, as illustrated in Figure 23.

D. AN UNKNOWN COMPONENT BETWEEN SEVERAL KNOWN COMPONENTS

If there is a symbol with unknown coordinates, $U_1(X_{U1}, Y_{U1})$ connected to n symbols with known coordinates, $K_1(X_1, Y_1)$ to $K_n(X_n, Y_n)$. As shown in Figure 24, the coordinates of the unknown point can be found using formula (4).

$$U1(X_{U1}, Y_{U1}) = \left(\frac{\sum_{i=1}^n X_i}{n}, \frac{\sum_{i=1}^n Y_i}{n} \right) \quad (4)$$

APPENDIX B. BALAS MAPPING RULES

Table 7 presents examples of Balas-specific mapping rules. The rules act as interpreters between the P&ID and the simulator. They define how a process component (tank, pump, reactor) present in the P&ID will be described in the selected simulator, which simulator modules are used to describe the component.

ACKNOWLEDGMENT

As a thank you, the authors want to express their appreciation to Eemeli Hytönen, Research and Development Manager of Metsä Spring Oy and a Former VTT Technology Manager, who generously shared his knowledge with them over the course of this research.

REFERENCES

- W. de Paula Ferreira, F. Armellini, and L. A. De Santa-Eulalia, "Simulation in industry 4.0: A state-of-the-art review," *Comput. Ind. Eng.*, vol. 149, Nov. 2020, Art. no. 106868, doi: [10.1016/j.cie.2020.106868](https://doi.org/10.1016/j.cie.2020.106868).
- S. Boschert and R. Rosen, "Digital twin—The simulation aspect," in *Mechatronic Futures*. Cham, Switzerland: Springer, 2016, pp. 59–74.
- W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018, doi: [10.1016/j.ifacol.2018.08.474](https://doi.org/10.1016/j.ifacol.2018.08.474).
- A. Koulouris, N. Misailidis, and D. Petrides, "Applications of process and digital twin models for production simulation and scheduling in the manufacturing of food ingredients and products," *Food Bioprocess Process.*, vol. 126, pp. 317–333, Mar. 2021, doi: [10.1016/j.fbp.2021.01.016](https://doi.org/10.1016/j.fbp.2021.01.016).
- S. Haag and R. Anderl, "Digital twin—Proof of concept," *Manuf. Lett.*, vol. 15, pp. 64–66, Jan. 2018, doi: [10.1016/j.mfglet.2018.02.006](https://doi.org/10.1016/j.mfglet.2018.02.006).
- F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2018.
- J. Stjepandić, M. Sommer, and S. Stobrawa, "Digital twin: A conceptual view," in *DigiTwin: An Approach for Production Process Optimization in a Built Environment*. Cham, Switzerland: Springer, 2022, pp. 31–49.
- S. Sierla, M. Azangoo, K. Rainio, N. Papakonstantinou, A. Fay, P. Honkamaa, and V. Vyatkin, "Roadmap to semi-automatic generation of digital twins for brownfield process plants," *J. Ind. Inf. Integr.*, vol. 27, May 2022, Art. no. 100282, doi: [10.1016/j.jii.2021.100282](https://doi.org/10.1016/j.jii.2021.100282).
- S. Sierla, L. Sorsamäki, M. Azangoo, A. Villberg, E. Hytönen, and V. Vyatkin, "Towards semi-automatic generation of a steady state digital twin of a brownfield process plant," *Appl. Sci.*, vol. 10, no. 19, p. 6959, Oct. 2020.
- K. Vijayakumar, C. Dhanasekaran, R. Pugazhenthii, and S. Sivaganesan, "Digital twin for factory system simulation," *Int. J. Recent Technol. Eng.*, vol. 8, no. 1, pp. 63–68, 2019.
- A. Bamberg, L. Urbas, S. Bröcker, M. Bortz, and N. Kockmann, "The digital twin—Your ingenious companion for process engineering and smart production," *Chem. Eng. Technol.*, vol. 44, no. 6, pp. 954–961, Jun. 2021, doi: [10.1002/ceat.202000562](https://doi.org/10.1002/ceat.202000562).
- M. Todorović, D. Ivanišević, M. Vještica, V. Dimitrieski, and I. Luković, "An automatic generation of production documentation from MultiProLan models," in *Proc. 11th Int. Conf. Inf. Soc. Technol.*, 2021, pp. 96–101.
- E. Ors, R. Schmidt, M. Mighani, and M. Shalaby, "A conceptual framework for AI-based operational digital twin in chemical process engineering," in *Proc. IEEE Int. Conf. Eng., Technol. Innov. (ICE/ITMC)*, Jun. 2020, pp. 1–8, doi: [10.1109/ICE/ITMC49519.2020.9198575](https://doi.org/10.1109/ICE/ITMC49519.2020.9198575).
- R. Rosen, G. Von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-Papers OnLine*, vol. 48, no. 3, pp. 567–572, 2015, doi: [10.1016/j.ifacol.2015.06.141](https://doi.org/10.1016/j.ifacol.2015.06.141).
- G. S. Martinez, T. A. Karhela, R. J. Ruusu, S. A. Sierla, and V. Vyatkin, "An integrated implementation methodology of a lifecycle-wide tracking simulation architecture," *IEEE Access*, vol. 6, pp. 15391–15407, 2018, doi: [10.1109/ACCESS.2018.2811845](https://doi.org/10.1109/ACCESS.2018.2811845).
- Y. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, "Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues," *Robot. Comput.-Integr. Manuf.*, vol. 61, Feb. 2020, Art. no. 101837, doi: [10.1016/j.rcim.2019.101837](https://doi.org/10.1016/j.rcim.2019.101837).
- J. de Beer and C. Depew, "The role of process engineering in the digital transformation," *Comput. Chem. Eng.*, vol. 154, Nov. 2021, Art. no. 107423, doi: [10.1016/j.compchemeng.2021.107423](https://doi.org/10.1016/j.compchemeng.2021.107423).
- L. Sorsamäki, A. Koponen, and E. Hytönen, "Process simulation-based evaluation of design and operational implications of water-laid paper machine conversion to foam technology," *BioResources*, vol. 16, no. 3, p. 5148, 2021.
- V. Bavane, M. Studies, and R. V. Marode, "Digital twin?: Manufacturing excellence through virtual factory digital twin?: Manufacturing excellence through virtual," *Global J. Eng. Sci. Researches*, pp. 6–15, Nov. 2018. [Online]. Available: <https://zenodo.org/record/1493930#YpXh2ExRWbg>
- B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167653–167671, 2019, doi: [10.1109/ACCESS.2019.2953499](https://doi.org/10.1109/ACCESS.2019.2953499).
- B. S. Carlberg, "The autonomous mill: Utilizing digital twins to optimize the pulp & paper mill of the future," in *Proc. IEEE IAS Pulp Paper Ind. Conf. (PPIC)*, Jun. 2021, pp. 1–10, doi: [10.1109/PPIC47846.2021.9620318](https://doi.org/10.1109/PPIC47846.2021.9620318).
- I. A. Udugama, P. C. Lopez, C. L. Gargalo, X. Li, C. Bayer, and K. V. Gernaey, "Digital twin in biomanufacturing: Challenges and opportunities towards its implementation," *Syst. Microbiol. Biomanufacturing*, vol. 1, no. 3, pp. 257–274, 2021.
- ARC Advisory Group. *Creating and Deploying Digital Twins in the Process Industries*. Accessed: Feb. 28, 2022. [Online]. Available: <https://www.arcweb.com/blog/creating-deploying-digital-twins-process-industries>
- R. Hofmann, V. Halmschlager, S. Knöttner, B. Leitner, D. Pernsteiner, L. Prendl, C. Sejkora, G. Steindl, and A. Traupmann, "Digitalization in industry: An Austrian perspective," Climate and Energy Fund, Vienna, Austria, Tech. Rep., 2020, p. 122. [Online]. Available: www.klimafonds.gv.at and <https://energieforschung.at/wp-content/uploads/sites/11/2020/12/White-Paper-Digitalization-in-Industry.pdf>
- C. Appl, A. Moser, F. Baganz, and V. C. Hass, "Digital twins for bioprocess control strategy development and realisation," in *Advances in Biochemical Engineering/Biotechnology*, vol. 177, C. Herwig, R. Pörtner, and J. Möller, Eds. Cham, Switzerland: Springer, 2021, pp. 63–94.
- J. Stjepandić and M. Sommer, "Object recognition methods in a built environment," in *DigiTwin: An Approach for Production Process Optimization in a Built Environment*. Cham, Switzerland: Springer, 2022, pp. 103–134. [Online]. Available: https://doi.org/10.1007/978-3-030-77539-1_6.
- Tesseract-OCR. GitHub. Accessed: Feb. 28, 2022. [Online]. Available: <https://github.com/tesseract-ocr>
- E. S. Yu, J. M. Cha, T. Lee, J. Kim, and D. Mun, "Features recognition from piping and instrumentation diagrams in image format using a deep learning network," *Energies*, vol. 12, no. 23, p. 4425, Nov. 2019, doi: [10.3390/en1234425](https://doi.org/10.3390/en1234425).
- M. Gada, "Object detection for P&ID images using various deep learning techniques," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, Jan. 2021, pp. 1–5, doi: [10.1109/ICCCI50826.2021.9402386](https://doi.org/10.1109/ICCCI50826.2021.9402386).
- S.-O. Kang, E.-B. Lee, and H.-K. Baek, "A digitization and conversion tool for imaged drawings to intelligent piping and instrumentation diagrams (P&ID)," *Energies*, vol. 12, no. 13, p. 2593, Jul. 2019, doi: [10.3390/en12132593](https://doi.org/10.3390/en12132593).

- [31] S. Mani, M. A. Haddad, D. Constantini, W. Douhard, Q. Li, and L. Poirier, "Automatic digitization of engineering diagrams using deep learning and graph search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 673–679, doi: [10.1109/CVPRW50498.2020.00096](https://doi.org/10.1109/CVPRW50498.2020.00096).
- [32] R. Rahul, S. Paliwal, M. Sharma, and L. Vig, "Automatic information extraction from piping and instrumentation diagrams," in *Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods*, 2019, pp. 163–172, doi: [10.5220/0007376401630172](https://doi.org/10.5220/0007376401630172).
- [33] M. Sommer and K. Seiffert, "Scan methods and tools for reconstruction of built environments as basis for digital twins," in *DigiTwin: An Approach for Production Process Optimization in a Built Environment*. Cham, Switzerland: Springer, 2022, pp. 51–77. [Online]. Available: https://doi.org/10.1007/978-3-030-77539-1_4, doi: [10.1007/978-3-030-77539-1_4](https://doi.org/10.1007/978-3-030-77539-1_4).
- [34] M. Grau, W. Korol, J. Lützenberger, and J. Stjepandic, "Automated generation of a digital twin of a process plant by using 3D scan and artificial intelligence," in *Transdisciplinary Engineering for Resilience: Responding to System Disruptions*. Amsterdam, The Netherlands: IOS Press, 2021, pp. 93–102, doi: [10.3233/ATDE210087](https://doi.org/10.3233/ATDE210087).
- [35] G. S. Martinez, S. A. Sierla, T. A. Karhela, J. Lappalainen, and V. Vyatkin, "Automatic generation of a high-fidelity dynamic thermal-hydraulic process simulation model from a 3D plant model," *IEEE Access*, vol. 6, pp. 45217–45232, 2018, doi: [10.1109/ACCESS.2018.2865206](https://doi.org/10.1109/ACCESS.2018.2865206).
- [36] F. Stinner, M. Wiecek, M. Baranski, A. Kümpel, and D. Müller, "Automatic digital twin data model generation of building energy systems from piping and instrumentation diagrams," 2021, *arXiv:2108.13912*.
- [37] J. G. Campos, J. S. López, J. I. A. Quiroga, and A. M. E. Seoane, "Automatic generation of digital twin industrial system from a high level specification," *Proc. Manuf.*, vol. 38, pp. 1095–1102, Jan. 2019, doi: [10.1016/j.promfg.2020.01.197](https://doi.org/10.1016/j.promfg.2020.01.197).
- [38] G. Lugaresi and A. Matta, "Automated manufacturing system discovery and digital twin generation," *J. Manuf. Syst.*, vol. 59, pp. 51–66, Apr. 2021, doi: [10.1016/j.jmsy.2021.01.005](https://doi.org/10.1016/j.jmsy.2021.01.005).
- [39] J. Pawlewitz, A. Mankel, S. Jacquin, and N. Basile, "The digital twin in a Brownfield environment: How to manage dark data," presented at the Offshore Technol. Conf., Houston, Texas, USA, May 2020, Paper OTC-30537-MS. [Online]. Available: <https://doi.org/10.4043/30537-MS>, doi: [10.4043/30537-MS](https://doi.org/10.4043/30537-MS).
- [40] S. Sierla, M. Azangoo, A. Fay, V. Vyatkin, and N. Papakonstantinou, "Integrating 2D and 3D digital plant information towards automatic generation of digital twins," in *Proc. IEEE 29th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2020, pp. 460–467.
- [41] S. Sierla, M. Azangoo, and V. Vyatkin, "Generating an industrial process graph from 3D pipe routing information," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 85–92.
- [42] M. Azangoo, J. Salmi, I. Yrjola, J. Bensky, G. Santillan, N. Papakonstantinou, S. Sierla, and V. Vyatkin, "Hybrid digital twin for process industry using apros simulation environment," in *Proc. 26th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2021, pp. 1–4, doi: [10.1109/ETFA45728.2021.9613416](https://doi.org/10.1109/ETFA45728.2021.9613416).
- [43] E. Arroyo, M. Hoernicke, P. Rodríguez, and A. Fay, "Automatic derivation of qualitative plant simulation models from legacy piping and instrumentation diagrams," *Comput. Chem. Eng.*, vol. 92, pp. 112–132, Sep. 2016, doi: [10.1016/j.compchemeng.2016.04.040](https://doi.org/10.1016/j.compchemeng.2016.04.040).
- [44] H. Son, C. Kim, and C. Kim, "3D reconstruction of as-built industrial instrumentation models from laser-scan data and a 3D CAD database based on prior knowledge," *Autom. Construction*, vol. 49, pp. 193–200, Jan. 2015, doi: [10.1016/j.autcon.2014.08.007](https://doi.org/10.1016/j.autcon.2014.08.007).
- [45] M. Sommer, J. Stjepandic, S. Stobrawa, and M. V. Soden, "Automated generation of a digital twin of a manufacturing system by using scan and convolutional neural networks," in *Transdisciplinary Engineering for Complex Socio-technical Systems—Real-life Applications* (Advances in Transdisciplinary Engineering), vol. 12. Amsterdam, The Netherlands: IOS Press, 2020, pp. 363–372, doi: [10.3233/ATDE200095](https://doi.org/10.3233/ATDE200095).
- [46] C. Liu, L. Le Roux, C. Körner, O. Tabaste, F. Lacan, and S. Bigot, "Digital twin-enabled collaborative data management for metal additive manufacturing systems," *J. Manuf. Syst.*, vol. 62, pp. 857–874, Jan. 2022, doi: [10.1016/j.jmsy.2020.05.010](https://doi.org/10.1016/j.jmsy.2020.05.010).
- [47] S. Stobrawa, G. V. Münch, B. Denkena, and M.-A. Dittrich, "Design of simulation models," in *DigiTwin: An Approach for Production Process Optimization in a Built Environment*. Cham, Switzerland: Springer, 2022, pp. 181–204. [Online]. Available: https://doi.org/10.1007/978-3-030-77539-1_9, doi: [10.1007/978-3-030-77539-1_9](https://doi.org/10.1007/978-3-030-77539-1_9).
- [48] *Industrial Automation Systems and Integration—Product Data Representation and Exchange: AP231 (CD)—Process Design and Process Specifications of Major Equipment*, Standard ISO/CD 10303-231:1998(E), Geneva, Switzerland, 1998.
- [49] *Pidgraph: Digitalization of Brownfield Documents*. Bilfinger. Bilfinger DIGITAL NEXT GMBH. Accessed: Feb. 28, 2022. [Online]. Available: <https://digitalnext.bilfinger.com/solutions/pidgraph>
- [50] *Model Broker, Automatic Extraction of Information From Second Hand Design Material*. Semantum. Accessed: Feb. 28, 2022. [Online]. Available: <https://www.semantum.fi/products/modelbroker/>
- [51] (May 2021). *UniversalPlantViewer; See Your Industry in New Dimensions*. CAXperts. Caxperts GmbH. [Online]. Available: <https://www.caxperts.com/>
- [52] ARC Advisory Group. *Bentley's and Siemens' Vision for Cloud-based Distributed Engineering and Operations*. Accessed: Feb. 28, 2022. [Online]. Available: <https://www.arcweb.com/sites/default/files/Documents/client-sponsored/bentley-and-siemens-vision-for-cloud-based-distributed-engineering-and-operations.pdf>
- [53] S. Paliwal, A. Jain, M. Sharma, and L. Vig, "Digitize-PID: Automatic digitization of piping and instrumentation diagrams," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, May 2021, pp. 168–180.
- [54] E. Rica, S. Álvarez, and F. Serratos, "Group of components detection in engineering drawings based on graph matching," *Eng. Appl. Artif. Intell.*, vol. 104, Sep. 2021, Art. no. 104404, doi: [10.1016/j.engappai.2021.104404](https://doi.org/10.1016/j.engappai.2021.104404).



MOHAMMAD AZANGOO (Member, IEEE) received the Master of Science degree in electrical engineering from the K. N. Toosi University of Technology, in 2014. He is currently pursuing the Ph.D. degree with the IT in Automation Group, Aalto University. He worked in industry for many years and has experience with diverse industrial projects in a variety of sectors, including energy and power, manufacturing, and oil and gas. His research interests include industry 4.0, digital twins, industrial automation, cyber physical systems, and process control. His current research focuses on developing new solutions for automatically producing digital twins and industry 4.0-related technologies.



LOTTA SORSAMÄKI received the Master of Science (M.Sc.) degree in the field of chemical engineering from the Helsinki University of Technology (nowadays Aalto University), in 2007.

After graduating, she began her professional career as a Process Designer in an engineering company designing bio product and chemical plants. Since 2009, she has been working as a Research Scientist at the VTT Technical Research Centre of Finland. She has 15 years expertise in conceptual process design, process simulation and modeling as well as conducting techno-economic assessments (TEA) in the field of bioprocessing, food, chemical, and pulp and paper industry.



SEPPO A. SIERLA (Senior Member, IEEE) was born in 1977. He received the M.Sc. degree in technology with the major in embedded systems and the D.Sc. degree in technology from the Helsinki University of Technology, in 2003 and 2007, respectively, and the Docent degree in software design for industrial automation from the School of Electrical Engineering, Aalto University, Finland, in 2013.

From 2003 to 2011, he was a Research Scientist with Aalto University and the Helsinki University of Technology. Since 2011, he has been a University Lecturer with Aalto University. His research interests include applications of simulation, machine learning, and ICT technologies to the energy sector.

Dr. Sierla is the Vice-Chair of the IEEE Finland Section (2022–2023) and before that he was the Section Treasurer (2020–2021). He is the Finance Chair of several IEEE conferences INDIN 2019, MLSP 2020, ISGT 2021 Europe, ISIT 2022, and ISIE 2023. Since 2020, he Co-Chairs the “Industrial Digitalization, Digital Twins in Industrial Applications” Track in the IEEE INDIN Conference Series.



TEEMU MÄTÄSNIEMI received the M.Sc. degree in the field of technical physics from the Tampere University of Technology (nowadays Tampere University), in 1998.

He worked with the Tampere University of Technology, as an Assistant (1994–1996) and a Researcher (2008–2009), as a Teacher in physics with Valmennuskeskus, in 1998, and as a Researcher with Finntech Oy (later Licentia Ltd.) (1996–2001). Since 2001, he has been working

at the VTT Technical Research Centre of Finland. He is currently acting as a Senior Scientist. He has experience in simulation and analysis software development, project and product management. His research interests include information management, ontologies, software processes, system integrations, and process plant safety.



MIIA RANTALA received the Master of Science (M.Sc.) degree in the field of automation and electrical engineering from Aalto University, in 2018. After graduation, she worked as a Software Developer at Semantum Oy. Her research interests include software development, pattern recognition, and graph algorithms.



KARI RAINIO received the M.Sc. degree in the field of technical physics from the Teknillinen Korkeakoulu-Tekniska Högskolan, in 1988. He currently works at the Sustainable Products and Materials, VTT Technical Research Centre of Finland. He does research in computing in mathematics, natural science, engineering and medicine, computer graphics, and instrumentation engineering. His most recent publication is MEMS FPI-Based Smartphone Hyperspectral Imager.



VALERIY VYATKIN (Fellow, IEEE) received the Ph.D. degrees in Russia and Japan, in 1992 and 1999, respectively, and the Habilitation degree in Germany, in 2002.

He is currently on Joint Appointment as the Chaired Professor with the Luleå University of Technology, Luleå, Sweden, and a Full Professor with Aalto University, Helsinki, Finland. Previously, he was a Visiting Scholar with Cambridge University, Cambridge, U.K., and had permanent academic appointments with New Zealand, Germany, Japan, and Russia. His research interests include dependable distributed automation and industrial informatics, software engineering for industrial automation systems, artificial intelligence, distributed architectures, and multiagent systems applied in various industry sectors, including smart grid, material handling, building management systems, data centers, and reconfigurable manufacturing.

Dr. Vyatkin was a recipient of the Andrew P. Sage Award for the Best IEEE TRANSACTIONS Paper in 2012. He has been the Chair of the IEEE IES Technical Committee on Industrial Informatics since 2016 and the Vice-President of IES for Technical Activities for the term 2022–2023.

...