

---

This is an electronic reprint of the original article.  
This reprint may differ from the original in pagination and typographic detail.

Sierla, Seppo; Azangoo, Mohammad; Fay, Alexander; Vyatkin, Valeriy; Papakonstantinou, Nikolaos

## **Integrating 2D and 3D Digital Plant Information Towards Automatic Generation of Digital Twins**

*Published in:*  
Proceedings of the 29th IEEE International Symposium on Industrial Electronics, ISIE 2020

*DOI:*  
[10.1109/ISIE45063.2020.9152371](https://doi.org/10.1109/ISIE45063.2020.9152371)

Published: 01/06/2020

*Document Version*  
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

*Please cite the original version:*  
Sierla, S., Azangoo, M., Fay, A., Vyatkin, V., & Papakonstantinou, N. (2020). Integrating 2D and 3D Digital Plant Information Towards Automatic Generation of Digital Twins. In *Proceedings of the 29th IEEE International Symposium on Industrial Electronics, ISIE 2020* (pp. 460-467). Article 9152371 (Proceedings of the IEEE International Symposium on Industrial Electronics). IEEE. <https://doi.org/10.1109/ISIE45063.2020.9152371>

# Integrating 2D and 3D Digital Plant Information Towards Automatic Generation of Digital Twins

Seppo Sierla<sup>1</sup>, Mohammad Azangoo<sup>1</sup>, Alexander Fay<sup>2</sup>, Valeriy Vyatkin<sup>1,3</sup>, and Nikolaos Papakonstantinou<sup>4</sup>

<sup>1</sup>*Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland*

<sup>2</sup>*Department of Automation Engineering, Helmut Schmidt University, Hamburg, Germany*

<sup>3</sup>*Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden*

<sup>4</sup>*VTT Technical Research Centre of Finland Ltd, Espoo, Finland*

seppo.sierla@aalto.fi, mohammad.azangoo@aalto.fi, alexander.fay@hsu-hh.de, valeriy.vyatkin@aalto.fi, nikolaos.papakonstantinou@vtt.fi

**Abstract**—Ongoing standardization in Industry 4.0 supports tool vendor neutral representations of Piping and Instrumentation diagrams as well as 3D pipe routing. However, a complete digital plant model requires combining these two representations. 3D pipe routing information is essential for building any accurate first-principles process simulation model. Piping and instrumentation diagrams are the primary source for control loops. In order to automatically integrate these information sources to a unified digital plant model, it is necessary to develop algorithms for identifying corresponding elements such as tanks and pumps from piping and instrumentation diagrams and 3D CAD models. One approach is to raise these two information sources to a common level of abstraction and to match them at this level of abstraction. Graph matching is a potential technique for this purpose. This article focuses on automatic generation of the graphs as a prerequisite to graph matching. Algorithms for this purpose are proposed and validated with a case study. The paper concludes with a discussion of further research needed to reprocess the generated graphs in order to enable effective matching.

**Index Terms**—industry 4.0, process industry, digitisation, automation, modelling and simulation, digital twins, graph matching, digital plant, plant design.

## I. INTRODUCTION

A major goal of Industry 4.0 is to make plant information available to humans and machines throughout the network of enterprises involved in designing, commissioning and operating the plant [1]. This information includes design information as well as information gathered during the operation of the plant, so a life cycle wide information management strategy and supporting tool chains are required [2]. However, plant design information still often resides in proprietary and tool specific formats. A few exceptions exist, such as the Proteus XML schema for P&ID (Piping & Instrumentation Diagram) diagram exchange, which is supported by several leading P&ID tool vendors [3], and the PCF (Piping Component File) format for 3D isometrics, supported by leading tool vendors such as Hexagon PPM, Autodesk, Alias and PTC Creo. However, these are solutions for exchanging a specific type of diagram between tools of different vendors. There is also a need to integrate the information produced with different types of tools. The scope of this article is 2D information from P&IDs and 3D information from CADs, in the Proteus XML

and PCF formats, respectively. Our use cases for integrating such information is the generation of a digital twin, extending recent work [4] by combining the control loop information from the P&ID with the physical layout from the 3D CAD. A straightforward approach for information integration would be to match tag names from the 2D and 3D information sources to identify the parts of these models that correspond to the same component. However, with industrial design repositories it cannot be assumed that consistent naming conventions have been enforced to enable this approach [5]. Thus, the integration of 2D and 3D plant information is a challenging task, so it is helpful to break it down to a process consisting of several steps. Further research on proposing the steps of such a process is solicited from other research groups. In this paper, the following process is suggested:

- 1) Digitize the information to a standard, industrially accepted Industry 4.0 format. This may involve no effort if the designs were made in tools that support these formats. However, industrial plants have lifecycles of several decades, in which case innovative applications are required to digitalize the legacy design information. Such work has been done for P&IDs [6]–[8].
- 2) Raise the level of abstraction of the 2D and 3D designs, so that they are at the same level of abstraction.
- 3) Match the models generated in step 2, to identify the elements in these models that correspond to the same plant component, such as a tank or pump.
- 4) Use the matches to augment applications relying only on either 2D or 3D information sources. For example, [9] generate the physical aspect of a digital twin of a process based solely on the 3D information, so control software is not generated and a legacy control system is expected to be integrated as in [10]. [3] generate the cyber aspect of a digital twin, i.e. a control system, based on the 2D information. If the instrumentation in the 2D and 3D sources could be matched, it would be possible to automatically identify and connect the I/O interface of the physical and virtual aspects of the digital twin, eventually aiming at automatic generation of a system that could be considered a fully-fledged digital twin [11].

In this paper, it is expected that step 1 has been performed. The research goal of this paper is step 2. Steps 3 and 4 are presented for motivational purposes and they are left for further research.

## II. RELATED WORK

Significant prior work has been done with respect to step 1 of the process proposed in section I. Legacy engineering documents can be digitized by scanning and storing [12]. OCR (Optical Character Recognition) can be used to identify e.g. tag names in scans and, thus, to link engineering documents (such as equipment data sheets, work instructions, and operating manuals) that are related to each other. However, graphical data cannot be transformed into information, and links between items on a drawing cannot be transformed into a digital structural model. Several authors have investigated the extraction of text annotations from mixed text-graphic documents. [13] proposed a method for string separation in images with annotations. [14] introduced a raster-based method for the identification of string boxes. [15] proposed a hybrid algorithm for the same task. [16] presented a method for the recognition of both text and basic parametrical forms in documents. [17], [18] addressed the recognition of text from drawings. In [19], authors presented approaches for detection and segmentation of complex engineering drawings consisting of textual and graphical elements, aiming at identification of key elements only. Also, they published a comprehensive survey on alternative approaches for the digitisation of complex engineering drawings [20]. Other works have focused on the analysis of symbols (OSR), which is relevant e.g. in mechanical engineering to interpret and convert design drawings [21]. The ultimate goal is to generate 2D or 3D models in a neutral format. [22] presented a system which is able to interpret a range of engineering documents, such as logical diagrams, electrical circuits, and P&IDs. This approach does not support key geometric features such as scaling, rotation, and partial overlap of objects. [23] presented a method to analyse design drawings, esp. electric wiring diagrams. [24] proposed the combination of geometric and semantic information for the reconstruction of 3D CAD models from engineering drawings. The semantic information used in this approach is, however, limited to the recognition of symbols and does not consider semantic properties of the analyzed structural items. In addition, commercial methods exist which allow for automatic conversion of CAD designs into object-oriented models [25], but this requires access to the original CAD model software and can therefore not be applied to the typical use case where a plant owner has to rely on PDF documents. In [6], a method is described which combines OSR with semantic knowledge. This method allows extracting a structural model from a given 2D diagram, e.g. a P&I diagram or a control logic diagram. Furthermore, a method is described which merges the 2D P&I Diagram and the 2D control logic diagram into a single structural model. The method has been applied successfully to interpret engineering documents from an oil rig in the North Sea [26]. The method is limited as

it relies on a consistent, common naming scheme of the tag names in both diagrams. In [27], a P&I diagram is analysed for design faults based on the identified objects and their connections. A similar approach has been patented recently by T. Tung [28].

For step 2 of the process proposed in section I, several authors have identified graph formats as a suitable abstraction of complex engineering drawings. In [29], it has been described how to formulate rules which can be applied to convert structural plant models into more abstract models. For example, a P&ID which contains tanks, nozzles, pipes and joints can be converted into a structural model which provides all possible flow paths between a given set of tanks. [30] presents an application for the automatic generation of bond graph models from an IEC 62424 hierarchical representation of the process plants. Also, [5] convert 3D pulp&paper plant designs to graphs in order to perform graph matching to identify similar, and thus reusable designs. [31] has presented an approach for extracting information from P&ID sheets by using deep learning networks and low-level image processing techniques for capturing inlets, outlets and pipelines as a tree-like data structure. [32] uses graph abstractions to identify differences between process designs, as captured in 3D CAD models, and the as-built version of the plant, as captured by laser scans.

In recent years, many efforts have been made to standardize process presentation formats. The ISO 15926 standard information model with its Proteus XML file format [33], [34] focuses on the interoperability of P&IDs. A working group of owner operators, software vendors and research organizations called DEXPI developed a specification (DEXPI) based on the ISO 15926 to address practical issues and push for the adoption of the DEXPI/ISO15926 as an open P&ID storage format. DEXPI and OPC Foundation have formed a joint working group for defining a DEXPI OPC UA companion specification [35] to enable access of P&ID data over OPC UA communication platforms. Also, In [36], ISO 15926 and IEC 62424, i.e. two different standards for computer-accessible structural model descriptions, which have been conceived for the modeling of process plants, have been compared.

## III. CASE STUDY

The case study is a thermo-hydraulic water process (Fig. 1). The functionality of the process is not important for the aims of this article, but interested readers will find more details in [4], [9], [10], [37].

The process has been modelled in the Intergraph Smart 3D tool (Fig. 2), which is capable of exporting PCF files. The model includes 10 pipelines, each of which has its corresponding PCF file. Pipelines may have branches. The endpoint of a pipeline is either a nozzle of process equipment or an open endpoint, referencing another open end point in another PCF file.

A P&ID has been developed in the SmartPlant P&ID and exported with its ISO 15926 export tool. The exported file conforms to the Proteus 3.6 XML schema. Fig. 3 presents a



Fig. 1. Case process

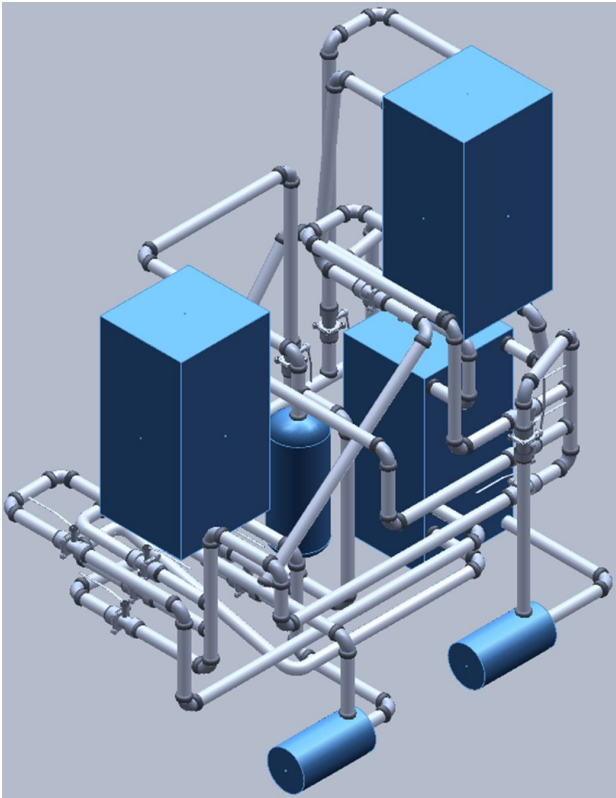


Fig. 2. 3D CAD model of the plant in Fig. 1.

visualization of the exported XML file. It is notable that Fig. 3 includes only the main pipelines, while Fig. 2 includes all of the pipelines. In general, such differences may be encountered in industrial plants, especially when working with design documents originating from different phases of the plant life-cycle.

#### IV. METHODOLOGY

Directed graphs with node labels are chosen as the abstraction level for step 2 of the procedure presented in Section I, as it is anticipated that they can support the matching activity in step 3. Such graphs have been previously successfully applied to matching industrial process plant design [5]. Matching of P&IDs and 3D models has not yet been attempted. The research goal stated in Section I can thus be elaborated as follows: to generate directed graphs with relevant node labels from P&IDs in Proteus DEXPI format and 3D CAD models in PFC format. Since the goal is to raise the level of abstraction, it is intended that the graph capture only a part of the information in the source document. The ideal level of information to be captured depends on the needs of steps 3 and 4 of the procedure introduced in Section I, so this is a discussion that is initiated in this paper and continued in further research. However, previous research on graph matching has shown that performance has been improved by graph simplification methods that have discarded details related to piping [5], so our starting point in this paper is that more detail is not necessarily better. The graph is specified as a set of node  $N$  and a set of directed edges  $E$ . Each edge is specified by source and target nodes  $e_{source}$  and  $e_{target}$ , which are elements of  $N$ .

##### A. A Generating a graph from a Proteus XML file

Fig. 4 presents a flowchart for generating the graph from an XML file conforming to the Proteus XML schema. The procedure extracts the connections between elements of the physical process or the control system, as opposed to graphical connections in the diagram. The `< Connection >` element of `< PipingNetworkSegment >` elements specifies connections nozzles of tanks or pumps to each other. In some cases, a `< PipingNetworkSegment >` connects to a valve and in some cases the valve is skipped, in the sense that the piping network segments have no information to specify that a valve was along that segment. Whether this occurs depends on the way in which the engineer uses the P&ID tool. For the control system, connectivity is specified in terms of the `¡Connect¡` elements of `< SignalLine >` elements. However, it was discovered that these connections are between two elements of type `< InstrumentComponent >`, which in our case are valves, heating elements, pump motors or generic actuators of unspecified type. Thus, this would result in many small stand-alone graphs not connected to the graph generated from `< PipingNetworkSegments >`. For example there is a `< SignalLine >` between the temperature sensor TI-T100 to the heating element ES-E100 (see tank B-100 in Fig. 3), but ES-E100 is not logically connected to the tank; it is just drawn next to the tank so that a human will understand that it refers

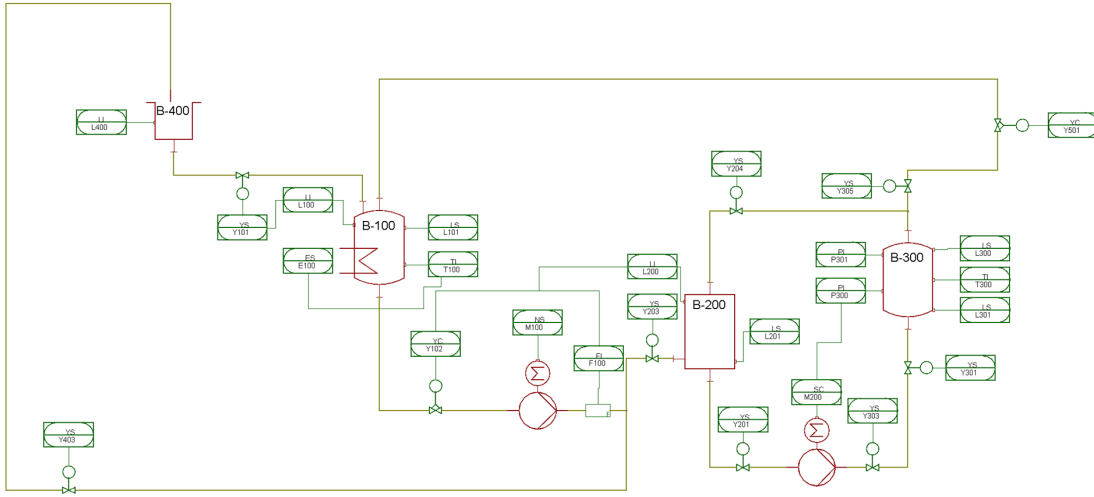


Fig. 3. P&ID of the plant in Fig. 1.

to the heating element in the tank. Thus, although extraction of *< InstrumentComponent >* and *< SignalLine >* elements was implemented, it was concluded after examining the results for the case study that it is very questionable whether these would add value to the generated graph. Thus, the procedure in Fig. 4 does not examine these elements. It is understood that further research is needed to determine the ideal level of detail for graphs generated from Proteus XML.

The flowchart in Fig. 4 attaches 3 kinds of labels to nodes:  $n_{name}$  (a unique id),  $n_{label}$  (a tag for human readable presentation) and  $n_{class}$  (which specifies the type of component and may be used later for graph matching purposes).

#### B. Generating a graph from a PCF file

Fig. 5 presents a procedure for generating a directed graph from a PCF file. The ‘New Component?’ element of the MAIN ALGORITHM examines components of type PIPE, WELD and VALVE. The PCF also defines TEE-STUB elements, but the branches in the pipelines can be captured in the graph without examining these elements. Each component has two END-POINT lines in the PCF file, which specify 3D coordinates. Each such coordinate will result in a node in the graph. The two END-POINTS are used to define an edge between the nodes that they define. The edge is labelled with the type of component; the types relevant for our case study are ‘Pipe’, ‘Weld’ and ‘Valve’. Thus, these nodes do not correspond to nodes in the graph generated from the P&ID. ALGORITHM2 in Fig. 5 extracts end connections from the PCF and generates nodes for them as well. The end connection of a pipeline is either a nozzle of process equipment or an open endpoint, referencing another open endpoint in another PCF file. In the case of a nozzle of process equipment, the created node will have a direct correspondence to a node generated from the P&ID. The label of this node generated by ALGORITHM2 is a string that combines tag and component type information, thus merging information similar to  $n_{label}$  and  $n_{class}$  generated by the algorithm Fig. 4.

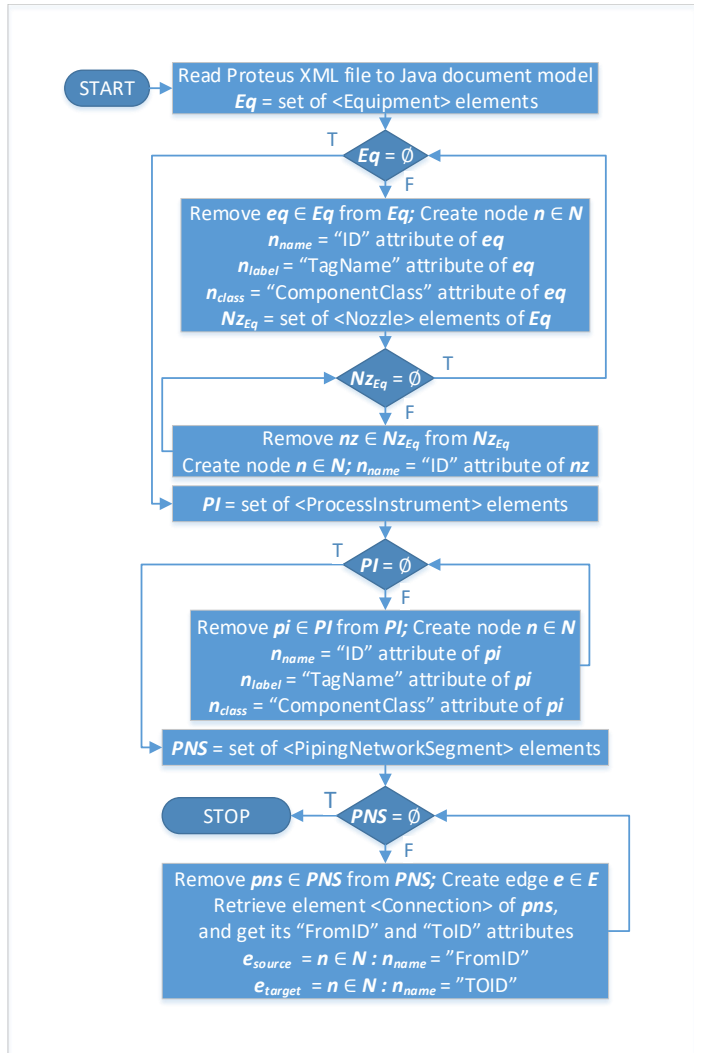


Fig. 4. Flowchart for generating a directed graph from an XML file conforming to Proteus XML schema.

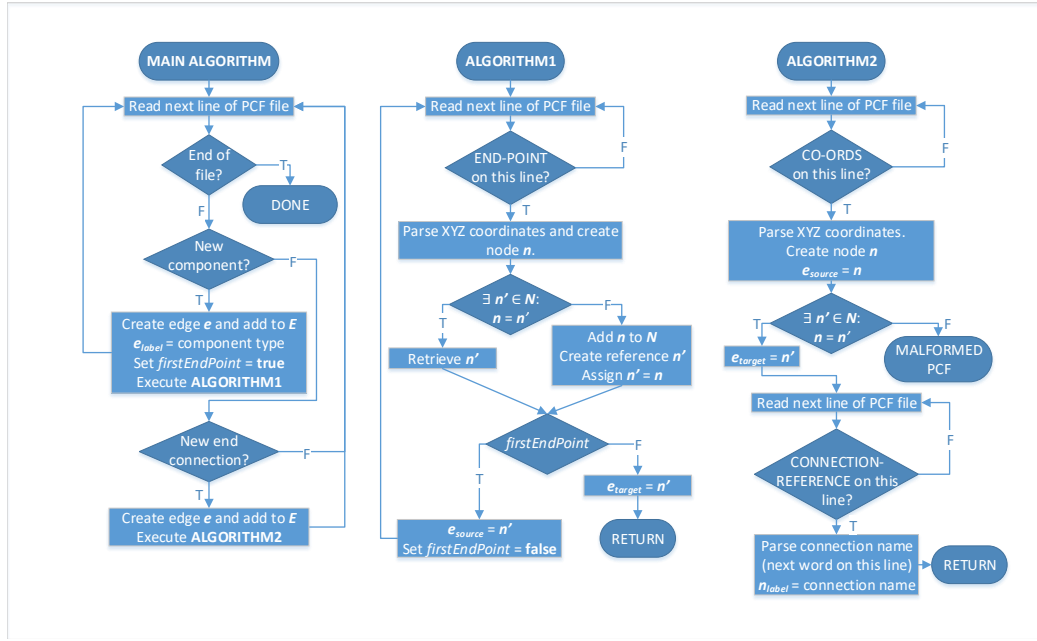


Fig. 5. Flowchart for generating a directed graph from a PCF file.

## V. RESULT

### A. Graph abstraction of the 2D model

The algorithm in Fig. 4 was implemented in Java and applied to a Proteus XML file corresponding to the P&ID in Fig. 3. Fig. 6 shows an excerpt of the generated graph, which was drawn manually from the list of nodes and edges exported by the algorithm. Fig. 7 shows the corresponding part of the P&ID. It is notable that in this case, the graph generation skips the inline instruments, the flow meter and the valve, for reasons explained in IV-A.

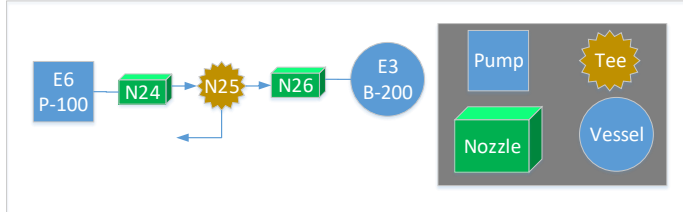


Fig. 6. Excerpt of generated graph.

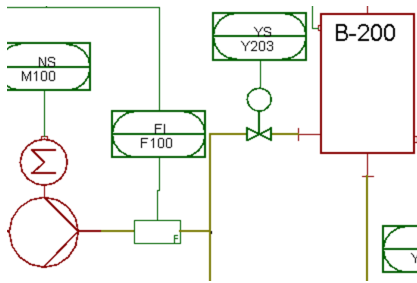


Fig. 7. Excerpt of P&ID corresponding to the graph in Fig. 6.

Fig. 8 shows the complete generated graph, which was drawn manually from the list of nodes and edges outputted by the algorithm. Referring to the node labels created by the algorithm in Fig. 4,  $n_{name}$  is a very long and hardly human readable id, which has been replaced in our implementation by a unique short human readable id such as E3, N16, or I5.  $n_{label}$  is a tag for human readable presentation and may not always be present in the XML file; in Fig. 8 it has been used in addition to the id (in case of tanks and pumps) and instead of the id in case of sensors.  $n_{class}$  provides the information on component types in the legend.

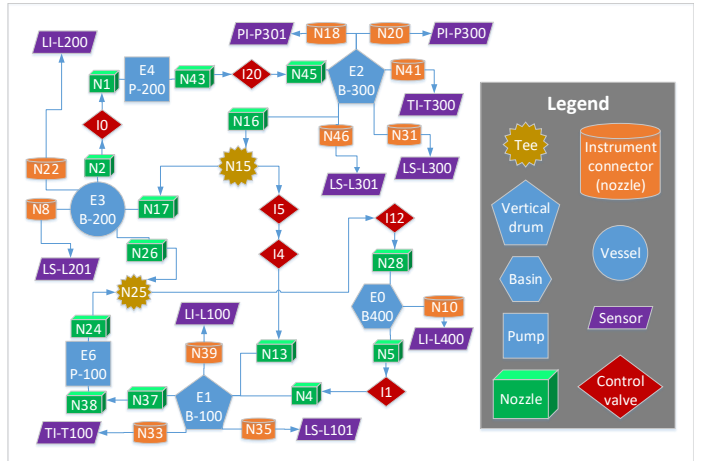


Fig. 8. Complete graph generated from P&ID.

### B. Graph abstraction of the 3D model

Fig. 9 shows the graph generated from the PCF of the simplest of the 10 pipelines in Fig. 2. The graph was drawn



manually from the list of nodes and edges outputted by the algorithm in Fig. 5. Color coding is used to show the corresponding process components on the photo of the pipeline.

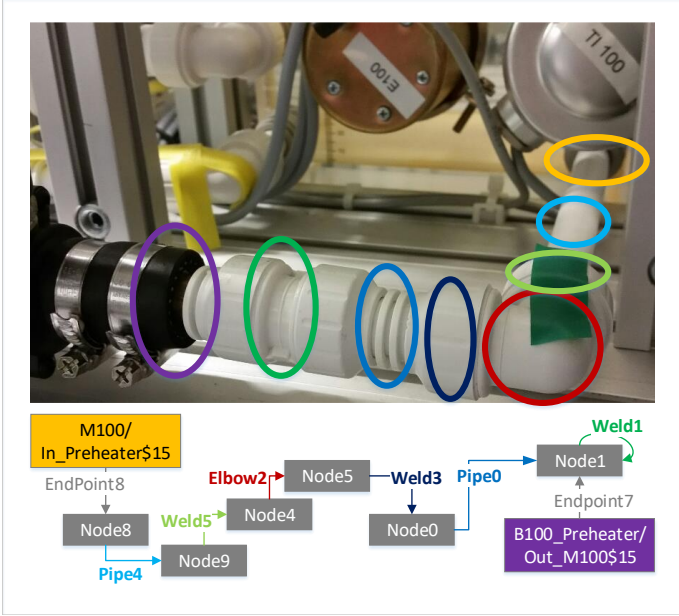


Fig. 9. Graph for the pipeline from tank 100 to pump 100

Fig. 10 repeats the experiment on a more complex pipeline with branches and valves, and Fig. 11 uses color coding to mark the corresponding elements on a photo of the pipeline.

It is notable that the edge directions do not correspond to flow directions, which may be a major issue for matching approaches based on directed graphs. Three possible solutions are proposed for further research:

- 1) It would be possible to examine the FLOW attributes of the PCF. However, these are optional attributes and it cannot be assumed that modelers define them, so this approach is not recommended.
- 2) The START-CO-ORDS attribute of the pipeline can be used. This can be used to identify the end connection node on the pipeline from which the flow originates. The solution depends on assuming that in 1 PCF file there are branches but no loops and that flows in all branches are away from the node at START-CO-ORDS. START-CO-ORDS is an optional attribute, but this assumption could be enforced by a semi-automatic solution that asks the users to specify the start node for each pipeline. If there is a usable interface which allows the user to select from options in a drop-down menu, the manual workload would be minimal. In this case, the edge directions can be fixed to correspond to flow directions by treating the graph generated from the PCF file as a tree with the node at START-CO-ORDS as the root node. The graph could then be processed with a tree traversal algorithm, so that edge directions are fixed to always point away from the root.

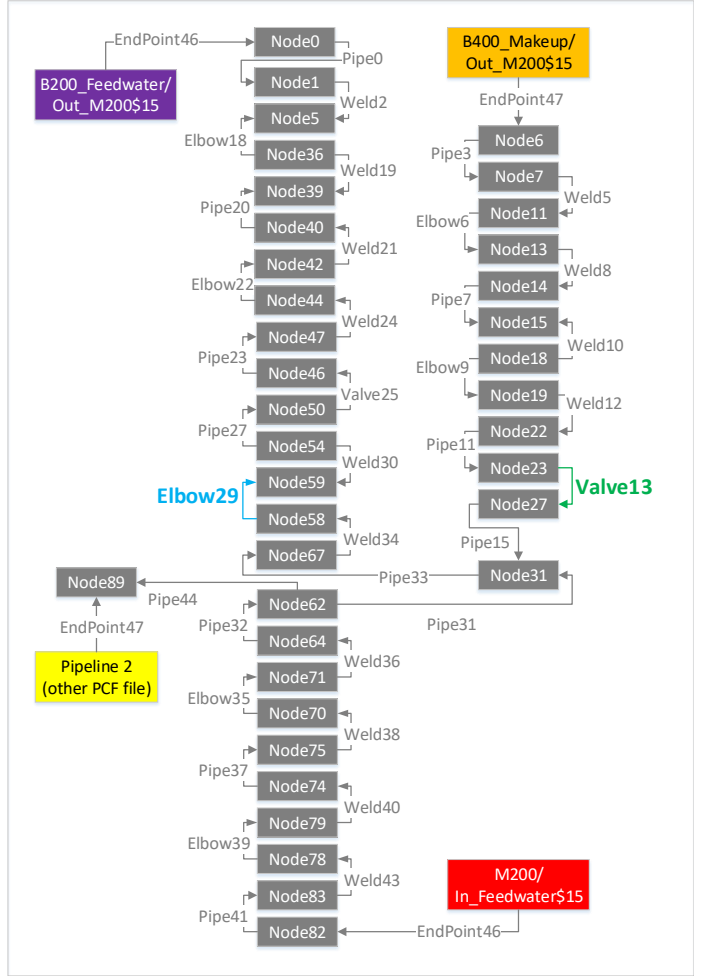


Fig. 10. Graph generated from the PCF of the pipeline from tank 200 to pump 200.

- 3) The ingoing and outgoing flows at pumps are specified in the end connection information of the PCF. In case of pipelines without pumps, in which the flow is caused by gravity, the elevations of the endpoints can be used to infer the flow direction. This could be used to overcome the need for manual input in solution 2 in case START-CO-ORDS has not been used.

## VI. DISCUSSION

Fig. 12 matches the PCF generated graph in Fig. 9 to the Proteus generated graph in Fig. 8. Color-coding is used to show the matching elements. The blue color makes it clear how the graphs are at a different level of abstraction. Graph simplification methods such as presented in [5] could readily be applied to eliminate this difference; however, the raw outputs are presented in Fig. 12, since the ideal graph simplification approach is a matter of further research.

Fig. 13 matches the PCF generated graph Fig. 10 to the Proteus generated graph in Fig. 8. It is notable that some parts of the PCF generated graph could not be matched, as they correspond to pipelines not included in the simplified



Fig. 11. Marking the color coded nodes and edges in Fig. 10 with colored ellipses on the relevant process components on the photo of the pipeline.

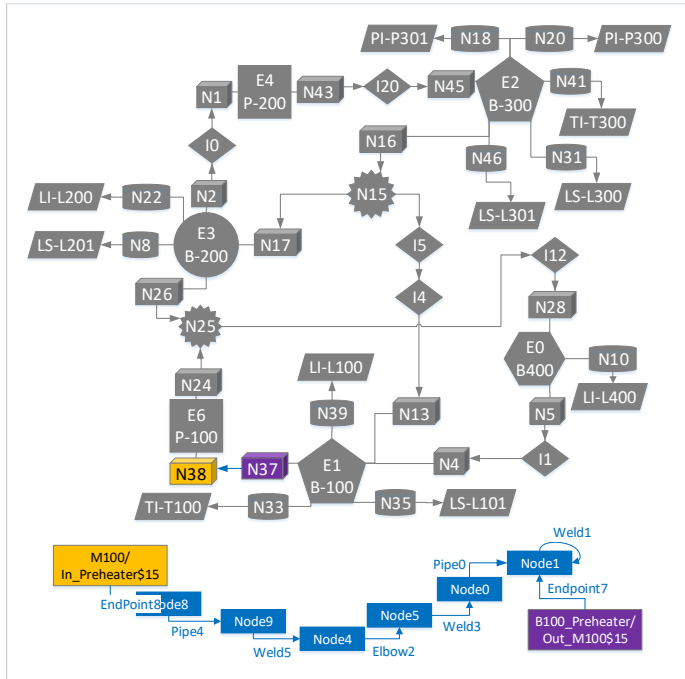


Fig. 12. Matching the graph in Fig. 9 to the graph in Fig. 8.

P&ID. As discussed in Section III, such a scenario is likely to occur in industrial practice over the plant lifecycle and solutions developed in further work should be robust against these scenarios.

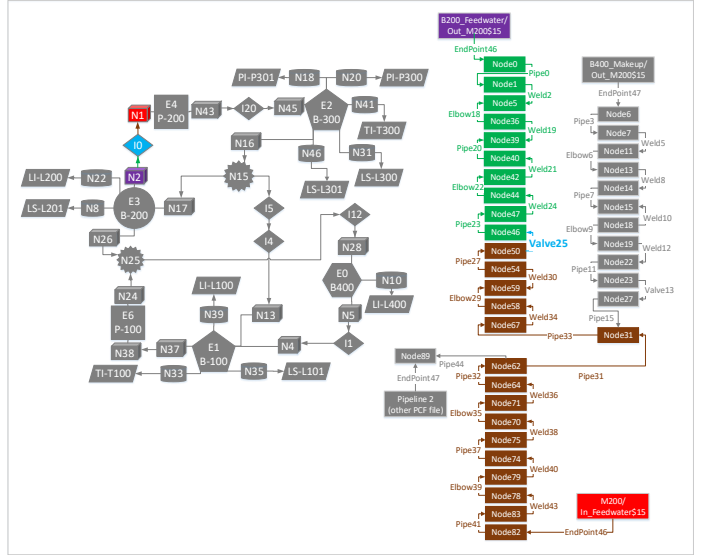


Fig. 13. Matching the graph in Fig. 10 to the graph in Fig. 8.

The color-coding in Fig. 12 and Fig. 13 was added manually. The automation of this matching work belongs to step 3 of the procedure introduced in Section I and is expected to be done in further work by graph matching techniques similar to [5]. It is notable that the graphs generated by the algorithms in Fig. 4 and Fig. 5 are a straightforward abstraction of the information in the source formats. Thus, they may not be ideal inputs for graph matching methods in further work. In particular, nodes in the graph generated from a P&ID correspond to process components and have a label  $n_{class}$ , which specifies the type of component. However, the PFC file specifies components such as pipe segments, welds and valves with result in edges. In other words, nodes in the P&ID graph may correspond to edges in the PCF graph (such as the indigo coded elements in Fig. 13).

## VII. CONCLUSION

To summarize the discussion, a preprocessing phase may be needed before graph matching to address the identified disparities between the graphs generated from the 2D and 3D sources. In particular, piping simplifications algorithms as in [5] could be applied to the graphs generated from the 3D CAD to arrive at the same level of details as in the P&ID graphs. Additional novel preprocessing algorithms are required to address disparities such as valves being represented as nodes in the 2D graph and as edges in the 3D graph. Finally, the findings suggest that level of tool support and industry standardization for capturing flow directions may be insufficient for the development of robust and general solutions for generating directed graphs from 3D CAD models. In this case, one viable option is to work with undirected



graphs, since according to previous research the direction information is only used to variants of the graph matching algorithm, such as the ‘anchor similarity measure’ in [5]. After these preprocessing steps, it is reasonable to expect the graph matching will give good results, since the graphs to be matched have similar structure and level of detail. The matching will provide the basis for integrating the 2D and 3D information to a single digital plant model.

## VIII. ACKNOWLEDGEMENTS

This work was partially supported by Business Finland project SEED (grant 4153/31/2019.)

## REFERENCES

- [1] F. Zezulka, P. Marcon, I. Vesely and O. Sajdl, “Industry 4.0 – An Introduction in the phenomenon,” IFAC-PapersOnLine, vol.49, issue 25, pp. 8-12, 2016.
- [2] R. Harrison, D. Vera and B. Ahmad, “Engineering Methods and Tools for Cyber-Physical Automation Systems,” in Proceedings of the IEEE, vol. 104, no. 5, pp. 973-985, May 2016.
- [3] N. Papakonstantinou, J. Karttunen, S. Sierla and V. Vyatkin, “Design to automation continuum for industrial processes: ISO 15926 – IEC 61131 versus an industrial case,” 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 2019, pp. 1207-1212.
- [4] G. S. Martínez, 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,” in IEEE Access, vol. 6, pp. 45217-45232, 2018.
- [5] M. Rantala, H. Niemistö, T. Karhela, S. Sierla and V. Vyatkin, “Applying graph matching techniques to enhance reuse of plant design information,” Computers in Industry, vol. 107, pp. 81-98, 2019.
- [6] E. Arroyo, M. Hoernicke, P. Rodríguez and A. Fay, “Automatic derivation of qualitative plant simulation models from legacy piping and instrumentation diagrams,” Computers in Chemical Engineering, vol.92, pp. 112-132, 2016.
- [7] M. Barth, A. Fay, “Automated generation of simulation models for control code tests,” Control Engineering Practice, vol.21, issue 2, pp. 218-230, 2013.
- [8] J. Nurminen, K. Rainio, J.-P. Numminen, T. Syrjänen, N. Paganus and K. Honkoila, “Object Detection in Design Diagrams with Machine Learning,” CORES 2019: The 11th International Conference on Computer Recognition Systems, Polanica-Zdroj, Poland, 2019.
- [9] G. S. Martínez, S. Sierla, T. Karhela and V. Vyatkin, “Automatic Generation of a Simulation-Based Digital Twin of an Industrial Process Plant,” IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, 2018, pp. 3084-3089.
- [10] G. S. Martínez, T. A. Karhela, R. J. Ruusu, S. A. Sierla and V. Vyatkin, “An Integrated Implementation Methodology of a Lifecycle-Wide Tracking Simulation Architecture,” in IEEE Access, vol. 6, pp. 15391-15407, 2018.
- [11] C. Koulamas and A. Kalogeras, “Cyber-Physical Systems and Digital Twins in the Industrial Internet of Things,” Computer, vol.51, issue 11, pp. 95-98, 2018.
- [12] “Viewport Operations.” viewport.ai. <https://viewport.ai/solutions/products/viewport-operations> (accessed Jan. 30, 2020).
- [13] S. Chowdhury, S. Mandal, A. Das and B. Chanda, “Segmentation of Text and Graphics from Document Images,” Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Parana, 2007, pp. 619-623.
- [14] J. Gao, L. Tang, W. Liu and Z. Tang, “Segmentation and recognition of dimension texts in engineering drawings,” Proceedings of 3rd International Conference on Document Analysis and Recognition, Montreal, Quebec, Canada, 1995, pp. 528-531 vol.1.
- [15] I. Chai and D. Dori, “Extraction of text boxes from engineering drawings,” Proc. SPIE 1661, Machine Vision Applications in Character Recognition and Industrial Inspection, August 1992.
- [16] L. Wenxin and D. Dori, “Genericity in graphics recognition algorithms,” Tombre K., Chhabra A.K. (eds) Graphics Recognition Algorithms and Systems. GREC 1997. Lecture Notes in Computer Science, vol. 1389. Springer, Berlin, Heidelberg, 1998.
- [17] L. Zhaoyang, “Detection of text regions from digital engineering drawings,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 4, pp. 431-439, April 1998.
- [18] C.-C. Han and K.-C. Fan, “Skeleton generation of engineering drawings via contour matching,” Pattern Recognition, vol. 27, issue 2, pp. 261-275, ISSN 0031-3203, 1994.
- [19] C. F. Moreno-García, E. Elyan and C. Jayne, “Heuristics-Based Detection to Improve Text/Graphics Segmentation in Complex Engineering Drawings,” Communications in Computer and Information Science, vol. 744, Springer, 2017.
- [20] C. F. Moreno-García, E. Elyan and C. Jayne, “New trends on digitisation of complex engineering drawings,” Neural Comput. & Applic., vol. 31, issue 6, pp. 1695-1712, 2019.
- [21] T. C. Henderson, “Analysis of Engineering Drawings and Raster Map Images,” Springer, 10.1007/978-1-4419-8167-7, 2014.
- [22] Y. Yu, A. Samal and S. C. Seth, “A system for recognizing a large class of engineering drawings,” Proceedings of 3rd International Conference on Document Analysis and Recognition, Montreal, Quebec, Canada, 1995, pp. 791-794 vol.2.
- [23] M. A. Berbar, “Automatic Diagrams Analysis,” Geometric Modeling and Imaging–New Trends (GMAI’06), pp. 160-170, 2006.
- [24] C. Ah-Soon and K. Tombre, “A step towards reconstruction of 3-D CAD models from engineering drawings,” Proceedings of 3rd International Conference on Document Analysis and Recognition, Montreal, Quebec, Canada, 1995, pp. 331-334 vol.1.
- [25] “xmplant.” <https://www.nextspace.nz>
- [26] M. Hoernicke, A. Fay and M. Barth, “Virtual plants for brown-field projects,” 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), Luxembourg, 2015, pp. 1-8.
- [27] W. C. Tan, I.-M. Chen, S. J. Pan and H. K. Tan, “Automated design evaluation on layout of Piping and Instrumentation Diagram using Histogram of Connectivity,” 2016 IEEE International Conference on Automation Science and Engineering (CASE), Fort Worth, TX, 2016, pp. 1295-1300.
- [28] Piping and instrumentation planning and maintenance system, by T. Tung, J.-I. Chatelain, J. A. Weichenberger, I. S. Grewal. (2018, Sep. 7). US10534983B1. Accessed on: Jan. 14, 2020. [Online]. Available: <https://www.patentguru.com/US10534983B1>
- [29] T. Schmidberger and A. Fay, “A rule format for industrial plant information reasoning,” 2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007), Patras, 2007, pp. 360-367.
- [30] S. Beez, A. Fay and N. Thornhill, “Automatic generation of bond graph models of process plants,” 2008 IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, 2008, pp. 1294-1301.
- [31] R. Rahul, S. Paliwal, M. Sharma and L. Vig, “Automatic Information Extraction from Piping and Instrumentation Diagrams,” arXiv preprint arXiv:1901.11383, 2019.
- [32] 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,” Automation in Construction, vol. 49, part B, pp. 193-200, 2015.
- [33] DEXPI. “Data Exchange in the Process Industry (DEXPI).” 2019, from <http://www.dexpi.org/>.
- [34] Fiatech. “ISO 15926 Information Models and Proteus Mappings (IIMM) - Proteus schema.” Retrieved 2017, from <http://fiatech.org/information-management/projects/1161-iso-15926-information-models-and-proteus-mappings-iimm>.
- [35] M. Wiedau, L. von Wedel, H. Temmen, R. Welke and N. Papakonstantinou, “ENPRO Data Integration: Extending DEXPI Towards the Asset Lifecycle,” Chemie Ingenieur Technik, vol. 91, issue 3, pp. 240-255, 2019.
- [36] T. Holm, L. Christiansen, M. Göring, T. Jäger and A. Fay, “ISO 15926 vs. IEC 62424 — Comparison of plant structure modeling concepts,” Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), Krakow, 2012, pp. 1-8.
- [37] S. Sierla, T. A. Karhela and V. Vyatkin, “Automatic Generation of Pipelines Into a 3D Industrial Process Model,” IEEE Access, vol. 5, pp. 26591-26603, 2017.