



Engineer-To-Order Plant Design: Assessing System Complexity and Hour Use Based on Directed Network Graphs

Bertram, Christian Alexander; Mueller, Georg Otto; Mortensen, Niels Henrik

Published in:
2020 Annual IEEE International Systems Conference (SysCon)

Link to article, DOI:
[10.1109/SysCon47679.2020.9275912](https://doi.org/10.1109/SysCon47679.2020.9275912)

Publication date:
2020

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Bertram, C. A., Mueller, G. O., & Mortensen, N. H. (2020). Engineer-To-Order Plant Design: Assessing System Complexity and Hour Use Based on Directed Network Graphs. In *2020 Annual IEEE International Systems Conference (SysCon)* IEEE. <https://doi.org/10.1109/SysCon47679.2020.9275912>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Engineer-To-Order Plant Design: Assessing System Complexity and Hour Use Based on Directed Network Graphs

Christian Alexander Bertram
Department of Mechanical Engineering
The Technical University of Denmark
Kgs. Lyngby, Denmark
<https://orcid.org/0000-0001-5481-0584>

Georg Otto Mueller
Department of Mechanical Engineering
The Technical University of Denmark
Kgs. Lyngby, Denmark
<https://orcid.org/0000-0001-6048-1297>

Niels Henrik Mortensen
Department of Mechanical Engineering
The Technical University of Denmark
Kgs. Lyngby, Denmark
<https://orcid.org/0000-0002-1886-5358>

Abstract — For an engineer-to-order case company that produces processing facilities, the complexity of conducted product designs is visualized with network graphs. The graphs represent the primary flow of material between the main functional components of the facilities. The network graphs exposes the design solution space of the analyzed products and the densities of the network graphs are used as measures of system complexities. The densities are compared to hour use in the corresponding projects to reveal the impact on hour use based on this complexity measure.

For a population of facilities with similar and sometimes identical build, the network graphs revealed varying combinations of components connectivity. Comparing graph types to hour use indicated that dense graphs increase the likelihood of spending more hours in the development project – mainly in the later project phases where the financial impact of increased hour use is typically worse. The results raise the questions, whether all solutions are equally profitable and how network graphs could aid design choices early in the projects, in order to limit additional late-phase risk.

Keywords — *Engineer-To-Order, ETO, Plant, Complexity, Design, Network Graph, Topology, Risk, Hour Quantities,*

I. INTRODUCTION

There are 915,103,765 ways to combine six of the normal eight-stud LEGO bricks [1]. If the number of bricks was variable and the bricks were instead complex engineered components, then it is safe to assume that the number of combinations for Engineer-To-Order (ETO) products are much higher. Assuming that all of these combinations are not equally efficient and some might perform better than others for the given purpose of the product, there must be a subset of optimal solutions. It is this subset of the solution space that this paper aims to reveal by condensing ETO products into network graphs.

When engineering companies work by the concept of Engineer-To-Order or Modify-To-Order it entails a great deal of customization and new development for each customer order. Over time, this adds more and more variety to the solution space of products in the company and complexity to the solutions [2]. When adding complexity to an already complex product, it might jeopardize the manageability and the overview of the portfolio. Not knowing what was built in previous projects makes it increasingly hard to compare to current design work. Increasing complexity might also jeopardize the functionality of systems as inter-dependencies and correlations within and beyond the system becomes difficult to master [2].

There is a need to mitigate and limit the expanse in the solution space and the design complexity, to avoid excessive

R&D and lifecycle costs or delivery times [3] [4] [5]. For ETO companies and production businesses there have been much focus on portfolio rationalization such as standardization, modularization, platform design, complexity management and product configuration [6] [7]. A commonality between them is the need for a portfolio overview and support in navigating the design solution space.

Product portfolio management efforts and design support programs, aim to steer product creation towards the best possible outcomes. Whether that focuses on re-using proven designs (e.g. standardization), combining proven sub-solutions (e.g. modularization) or reducing the portfolio to a smaller and better solution space (e.g. rationalization), etc.

The aim of the study behind this paper, is to derive metrics based on higher level system information, for guiding product designers towards preferable designs. Preferable designs in this sense being functionally proven and with less financial risk. This particular paper describes the work of correlating the high level topology of ETO processing lines to the risk of overconsumption of project resources (i.e. project execution hours). The goal is to let product designers use this metric to indicate if simplification of the current design is needed to decrease the risk of overconsuming hours in later project phases.

The paper proceeds as follows. Section II introduces network graphs, how graph densities are calculated and how such graphs are used to estimate system complexity. Section III presents the methodology used in the described case study. Section IV presents the results from the casework – including graphs, graph density calculations and correlations with project hour use. Lastly, Section V captures the indications and variance observed in the results while relating it to project execution risk.

II. BACKGROUND

This section presents the state of the art research on which this study is based. Besides the three main areas of research, it is outline how this paper combines these fields anew.

A. Directed Network Graphs

A network is "any netlike or complex system or collection of interrelated things (...)" as defined by The Oxford English Dictionary [8] or "constituent components (or 'nodes') and the 'links' or interrelationships between them" [2]. The conceptual and mathematical network is the basis for "network-based analysis" and "graph theory". However these interconnected things and their connections can be

represented visually using *Network Graphs* [9]. These models can be further detailed as "Directed Graphical Models" where the connections are not only present or not, but also include information about the dominant direction [9].

B. Graph Density

When working with simple network graphs, a typical measure is the Graph Density [10] [11] [12]. It is a calculation of the ratio of the number of present links compared to the maximal number of possible links. In general the equations for density (d) of directed graphs (1) and un-directed (2) are given as:

$$d_{\text{directed}} = L / (N(N-1)) \quad (1)$$

$$d_{\text{un-directed}} = 2 \times d_{\text{directed}} \quad (2)$$

Where d is the density of the graph, L is the number of individual links present and N is the number of nodes. Fig 1 shows examples of both un-directed and directed network charts.

C. Graph Based Engineering Complexity

Reference [13] works with a structural complexity measure based on the topology of a system architecture. Their work focuses on the system elements and the connectivity between them and between systems. System complexity is summarized as *Matrix Energy* based on an adjacency matrix that captures the connectivity between the system elements. However, the work revolves around network graph visualizations of systems. They discuss these measures in relation to system modularity and development cost and indications for non-linear growth of cost with increasing complexity. They suggest a measure like this for system engineering decisions.

Reference [14] works with a structural complexity measure for complex engineering systems. It is based on three main factors; the complexities of the individual components that are included in the system, the complexities of the individual interactions (links) between components and lastly the topological complexities of the system i.e. how components are connected to one another in the system. The complexity measure is put to use through a network graph depicting components and their interactions within and across systems.

Reference [15] discusses complexity in a general sense – how complexity can be understood within many fields of interest; systems engineering, organizations, functions,

biology etc. The network graph is used to describe systems complexity as it incorporates both components and connections, based on the argument that both these aspects are important to assess complexity. The number and kinds of components included form the *compositional complexity* while the connectivity between them form the *relational complexity*. Together they form the *structural complexity*.

D. Using Network Graphs to Describe Complexity and Financial Risk

The connection between ETO products and complexity has been widely researched and discussed – namely the links between system complexity and integration difficulties. Likewise, a great deal of effort has gone into describing systems and complexities of systems network graphs. However, the combination of the abovementioned with the financial impact and risk related to system complexity, described through network graphs, has not yet been explored. Hence the existence of this study and paper to describe it.

III. METHODOLOGY

The methodology used for this study is described in this section. The following steps outlines the steps used in the industry case presented in IV. Industry Case Results.

A. Retrieve System Design Information and Identify System Topology

The high level system topology describes which key functional components are included in the solution and how their primary interaction (e.g. connections) are laid out. In order to obtain this, the key components which represent the bulk of products must be identified. This is done by reviewing product specific technical literature and interviewing experts. Project documentation (e.g. technical drawings and flowsheets) describing these and their layout must be retrieved for each reviewed product. Thus the topology of the systems are now identified. This is illustrated as the progression from (a) to (b) on Fig 2.

B. Formalize System Topology as Data and Produce Directed Network Graphs

The system topology information (inclusion of key components and their connectivity) is formalized as tabularized data. Each row of this dataset describes a link between two nodes in a specific product. Table 1 shows an example of this with numbering of products, links and letter-based categorization of individual components types. The network graphs in this paper are constructed using the programming language Python [16], however there are several other programming languages options available to do similar work. In the Python environment, two specific libraries are used; **Pandas** [17] for managing the data and **NetworkX** [18] for constructing the network graphs. As the graphs are produced, so is a collection of the metrics for each graph (nodes, links, calculated density). This is represented by the progression from (b) to (c) on Fig 2.

C. Calculate System Graph Density and Correlate to Project Resource Consumption

Based on the formalized system topology data, network graphs are created. They are constructed with nodes representing the key components and links representing

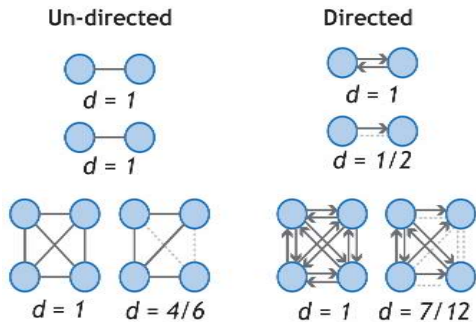


Fig 1: Examples of graphs and densities (d). Left column: Un-directed graphs. Right column: Directed graphs.

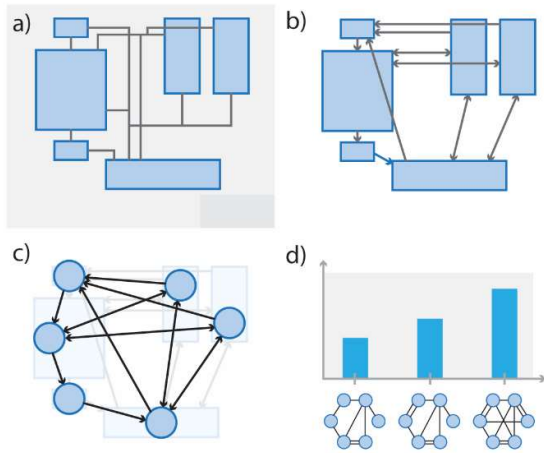


Fig 2: Transformation of system typology information into directed network graphs, following primary flow of material. (a) system documentation, (b) system topology, (c) directed network graph and (d) comparison of various directed network graphs

Table 1: Example of formalization of topology data

Product	Link	From	To
1	1	A	B
1	2	B	A
1	3	A	C
⋮	⋮	⋮	⋮

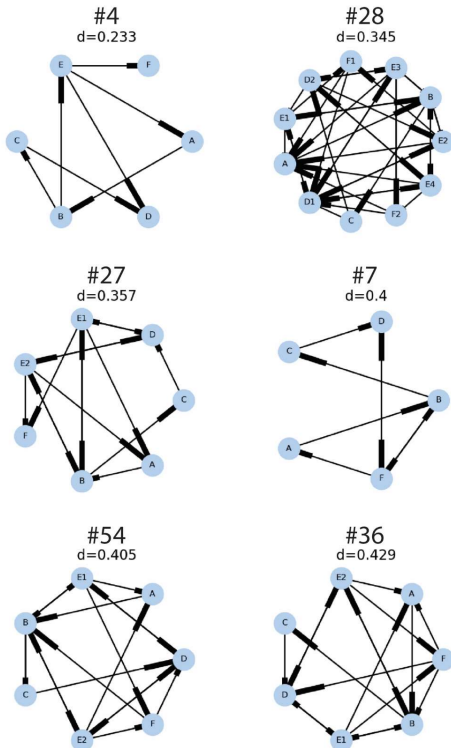


Fig 3: Six of the produced directed network graphs, with calculated graph densities in ascending order.

primary interactions (e.g. connections) between them. From these graphs, the densities are calculated using (1).

The graphs and the metrics describing them are then plotted against measures describing the execution of the respective projects, e.g. project execution hour consumption. This is illustrated on Fig 2 (d).

IV. INDUSTRY CASE RESULTS

This section presents the work done and results produced from applying this methodology on industry case products. The variation in graph topology is presented along with the variation in the calculated network graph densities as a measure for design complexity. Lastly, hour use in the design projects is plotted against the calculated graph densities to investigate possible correlations between design complexity and project execution effort.

A. Introduction to the Industry Case

The case for this work has been an ETO company which develops, designs, manufactures and erects large and complex processing facilities (plants). A subset of their design types has been the scope for this study. Across the case projects, the same types of key functional components are used. There are numerous other equipment and auxiliary functions, but this study has focused on the primary components which supplies the main functionality of the product. Within these component groups, there are different sub-types and sizes. However, the study was only concerned the inclusion/exclusion of component groups and their primary functional interaction with each other.

B. Retrieving Design Information and Identifying System Topology

For the products in scope of this study at the case company, the key functional components were identified partly from technical literature and verified with design engineers. For each of the processing plants built, there were documentation on the overall process flow through components. These flowsheets depicted the key functional components and the flow of material (i.e. powder product in these cases) through the processing plant. The topology of each of these plant designs was recorded in a database manually in accordance with Table 1.

C. From System Topology to Network Graphs

The key functional components were identified for each plant case and added as nodes to the network graphs. Each group of components are labeled A, B, C etc. If there were more entities present from the same group, they were given an additional number to the label, e.g. A1, A2 etc. The primary flow of product between these components was mapped onto the graphs as directed links.

Fig 3 presents six of the total 60 network graphs produced. The numbers (prefixed by #) correspond to the case plant number in Table 2. Please note that the direction of the graph links are marked with thickened line ends instead of arrowheads.

Table 2. Metrics of Nodes (N), Links (L) and Densities (D) of the produced network graphs.

#	Graph Metrics			#	Graph Metrics		
	N	L	d		N	L	d
#1	6	13	0.433	#31	6	10	0.333
#2	7	15	0.357	#32	5	7	0.350
#3	5	8	0.400	#33	5	7	0.350
#4	6	7	0.233	#34	6	11	0.367
#5	6	10	0.333	#35	7	18	0.429
#6	5	8	0.400	#36	7	15	0.357
#7	6	13	0.433	#37	7	18	0.429
#8	9	23	0.319	#38	6	10	0.333
#9	6	11	0.367	#39	6	10	0.333
#10	5	8	0.400	#40	8	18	0.321
#11	5	7	0.350	#41	7	15	0.357
#12	7	13	0.310	#42	6	9	0.300
#13	6	11	0.367	#43	7	15	0.357
#14	12	28	0.212	#44	6	9	0.300
#15	7	12	0.286	#45	7	15	0.357
#16	9	20	0.278	#46	9	23	0.319
#17	8	21	0.375	#47	9	23	0.319
#18	6	11	0.367	#48	6	8	0.267
#19	6	11	0.367	#49	10	23	0.256
#20	5	8	0.400	#50	5	7	0.350
#21	7	17	0.405	#51	6	8	0.267
#22	9	19	0.264	#52	5	8	0.400
#23	5	7	0.350	#53	7	17	0.405
#24	8	21	0.375	#54	9	28	0.389
#25	5	8	0.400	#55	6	11	0.367
#26	7	15	0.357	#56	6	9	0.300
#27	11	38	0.345	#57	7	13	0.310
#28	8	21	0.375	#58	6	11	0.367
#29	9	27	0.375	#59	7	18	0.429
#30	5	7	0.350	#60	7	13	0.310

D. Calculate Graph Densities and Compare to Hour Use in Project Execution

The density of each graph was calculated using (1) as described in section II.B. Table 2 shows all plants and the corresponding number of nodes (N) i.e. key components, number of links (L) i.e. primary connections and calculated density (d).

Besides visualizing the design solution space, this study aims at assessing the consequence of different designs on project execution effort. Hour allocation data has been available for the corresponding plant design projects. Together with company experts, these budget activities were divided into five categories:

- **Design:** Overall structure, flowsheets, physical arrangements, operational calculations etc.
- **Engineering:** Constructing and drawing components and plant elements.
- **Automation:** Outlining and programming the plant control software.
- **Commissioning:** Installing, starting and ensuring performance of the product on site.
- **General:** Project management, supply chain, support functions etc.

The first four categories correspond to project phases which are executed roughly in sequence, with some overlap. The last category (**General**) spans the entire project and is generally applicable to the whole project flow.

Fig 4 **Error! Reference source not found.** presents the hour use in the abovementioned five categories and how it correlate to the calculated graph densities. The five categories are presented twice, once with all case example plants and once with only the plant projects where all project activities have been reported fully completed. Linear trend lines have been added to the charts, to indicate the tendencies of hour use with increasing design graph density.

V. DISCUSSION

This section sums up the observations made from the case results, how they can be interpreted and what impact they might have.

A. Variance in Design

The network graphs shown in Fig 3 and the metrics in Table 2 indicate design variety. There are simple graphs with just five nodes and complex graphs with as much as 12 nodes, with the number of nodes corresponding to the number of key functional components in the design. When looking at the

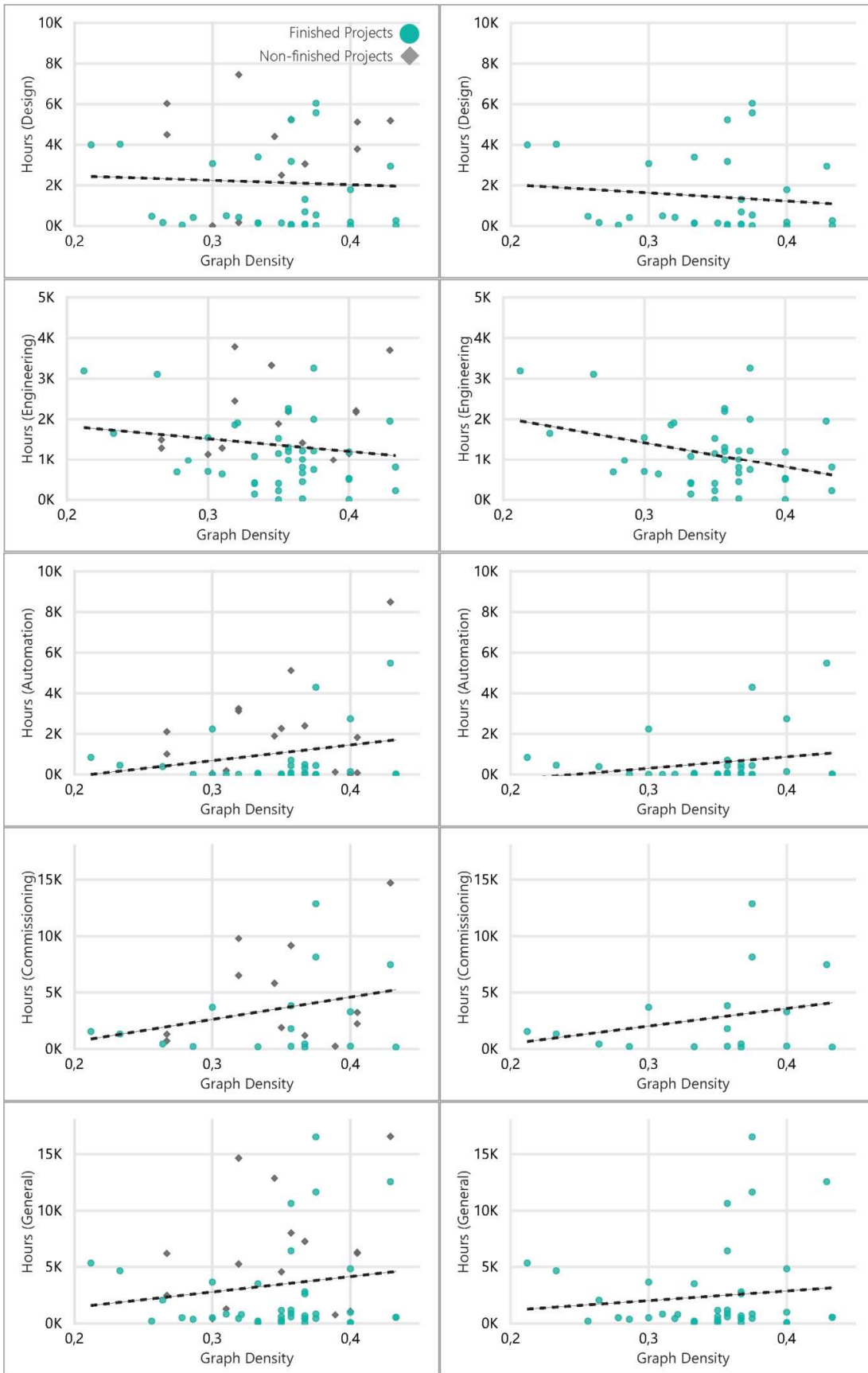


Fig 4: Hour use (consumption) in five categories of project execution. (Left column) all projects and (right column) only finished projects. Linear trend lines are added for correlation check.

many connections (38 links). The graph density are varying between 0.212 and 0.433. All of this indicates a varying design solution space, with different ways of combining components and connecting them.

B. Variance in Hour Use and Risk of Over-Consumption

Inspecting the charts in Fig 4 the linear correlations of the first two categories, **Design** and **Engineering**, are similar, showing steady or downward sloping trends. The third category, **Automation**, shows some upwards tendency for hour use as graph density increases. Even more noticeable are the tendencies for the last of the categories, **Commissioning** and **General**, as the linear regression shows that hour use in this phase tend to increase with increasing graph density.

When inspecting the charts shown in Fig. 3, it can be seen that high graph density values both correspond to low and high hour quantities. This is especially the case for the charts showing the hours in the categories **Commissioning** and **General**. It indicates an increased risk to use higher amounts of hours in these categories for projects that involve products with high graph density.

C. Financial Impact of Increased Hour Use

When building complex engineering products such as processing plants, which entail on-site activities with installation and start-up, there is a significant difference in the cost of hours spent, depending on the phase in which they are spent. Company experts stated that late stage hours and on-site hours are significantly more expensive, as they entail travel, accommodation, excess working hours and excess waiting times for material or personnel. Hence increased hour use in the later project phases has a greater financial impact than an equivalent increase in the early project phases.

D. Case-based Empiri

The study described in this paper supplies case-based evidence that the system topology of large-scale engineering products can be relevant to monitor and compare when doing ETO products. If higher project execution effort is expected from more complex product designs, then the presented results also indicates that network topology can be correlated to system complexity. This is important when doing early product design assessments.

E. Network Graphs as A Design Tool

Metrics like system topology density (Network Graph Density) can supply instant feedback to sales offices or product designers. Before deciding on a solution, they have the opportunity to evaluate potential correlations with project execution performance.

When developing, designing and constructing complex products, it is beneficial to know the consequences of the design choices being made and the consequential risk being added to the later project phases. Comparing current design work with previously conducted projects, in order to assess the complexity, the risk and possible financially safer alternatives, could prove valuable. The network graph and the calculated density as presented in this paper, could be calculated early in new plant design projects and be used to assess current design work before it is fully determined. Knowing that a certain design might add thousands of hours

in the later project stages could empower up-front design decisions.

F. Limitations of This Study

As with all metrics and tools that rely on high-level system information, it is crucial to understand that vital details might get lost. Thus such a support should be used mostly as guidance and less as a critical decision maker.

If utilizing graphing for other product types, it is important to define the proper system (and system elements) to include in the model. The authors recognize that further research should be done in this area, in order to quantify the consequences of changing systems and system boundaries for methods like this.

Further research is required to understand the correlation between other system metrics and project execution performance. There are many other factors to include and study to fully understand what makes a successful product.

VI. CONCLUSION

This paper presents a practical application of network graphs as a visualization tool and calculation basis for assessing the diversity and complexity of Engineer-To-Order (ETO) processing plant designs. Furthermore, it uses network graph density as an indicator for project execution risk – namely excessive use of hours in the later project stages.

A case population of 60 plant design projects from one case company was subject of the study. For each plant, key functional components and their connections were drawn as directed network graphs. Based on these graphs, the graph densities were calculated for each plant and used as a measure for design complexity. The impact of varying graph densities was shown by charting hour allocation in categorized project stages data against the varying network graph densities. The charts revealed that high values of network graph density (interpreted as complex designs) result in a higher risk of spending excessive quantities of hours in Automation (plant controls and software), **Commissioning** (installation and start-up) and **General** (project management).

Given the circumstances of the business, a higher amount of hours spent in the later project stages have a larger financial impact than the equivalent hour use increase in the early project stages. If designs that entail a risk of an increased hour use in the late project phases could be avoided and substituted with alternative designs of lower risk, potential project overruns could be avoided early on. The measure presented in this paper can be used to screen design projects against previous solutions to assess and possibly avoid the additional risk in the later project stages.

There are many other factors that affect the success and profitability of an ETO product. This paper is not intended to proclaim system complexity as the only dominant influencing factor. However, this paper indicates a link between high level system topology and the risk of an increased amount of hours spent in the later stages of the project.

The work presented in this paper will continue to explore other influencing factors on profitability in processing plant design and improve the presented network graph approach by incorporating more dimensions and design details.

ACKNOWLEDGMENT

The authors would like to thank the collaborating case company and its employees for great information access and data availability.

REFERENCES

- [1] Time, "The Number of Ways You Can Put Together 6 LEGO Bricks Will Astound You," 2015. [Online]. Available: <https://time.com/3977789/lego-brickumentary-math-professor-combinations/>. [Accessed 4 September 2019].
- [2] D. Fisk, "Engineering complexity," *Interdisciplinary Science Reviews*, vol. 29, no. 2, pp. 151-161, 2004.
- [3] K. Sinha and E. S. Suh, "Pareto-optimization of complex system architecture for structural complexity and modularity," *Research in Engineering Design*, vol. 29, no. 1, pp. 123-141, 2018.
- [4] A. Haug, L. Hvam and N. H. Mortensen, "Reducing variety in product solution spaces of engineer-to-order companies: The case of Novenco A/S," *International Journal of Product Development*, vol. 18, no. 6, 2013.
- [5] Z. Wang, M. Zhang, H. Sun and G. Zhu, "Effects of standardization and Innovation on mass customization: An empirical investigation," *Technovation*, Vols. 48-49, pp. 79-86, 2016.
- [6] C. Y. Baldwin and K. B. Clark, "Modularity in the Design of Complex Engineering Systems," in *Complex Engineering Systems*, Springer, 2006, pp. 175-205.
- [7] M. Gepp, M. Foehr and J. Vollmar, "Standardization , modularization and platform approaches in the engineer-to-order business - Review and Outlook," in 2016 Annual IEEE Systems Conference (SysCon), 2016.
- [8] "The Oxford English Dictionary," [Online]. Available: www.oed.com. [Accessed 4 September 2019].
- [9] E. D. Kolaczyk, *Statistical Analysis of Network Data – Methods and Models*, Springer, 2009.
- [10] T. F. Coleman and J. J. Moré, "(1983) "Estimation of Sparse Jacobian Matrices and Graph Coloring Problems," *Society for Industrial and Applied Mathematics*, vol. 20, no. 1, p. 187–209, 1983.
- [11] R. Diestel, *Graph Theory*, Springer, 2017.
- [12] P. E. Black, "Sparse Graph," *Dictionary of Algorithms and Data Structures* [online], 14 August 2008. [Online]. Available: <https://www.nist.gov/dads/HTML/sparsegraph.html>. [Accessed 12 September 2019].
- [13] K. Sinha and O. d. Weck, "Structural complexity quantification for engineered complex systems and implications on system architecture and design," in *Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE 2013*, Portland, Oregon, 2013.
- [14] K. Sinha, E. S. Suh and O. d. Weck, "Integrative Complexity: An Alternative Measure for System Modularity," *Journal of Mechanical Design*, vol. 140, no. 5, 2008.
- [15] G. E. Mobus and M. C. Kalton, "Principles of Systems Science," in *Complexity*, Springer, 2014, pp. 169-212.
- [16] Python Language Reference, version 3.6, available at <http://www.python.org>, Python Software Foundation.
- [17] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, 2010.
- [18] A. A. Hagberg, D. A. Schult and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, Aug 2008.