

This is a pre-print / author accepted version. Please cite article as:

Parraguez Ruiz, P., Piccolo, S., Perišić, M. M., Štorga, M., & Maier, A. (Accepted/In press). Process modularity over time: modelling process execution as an evolving activity network. *I E E E Transactions on Engineering Management*. <https://doi.org/10.1109/TEM.2019.2935932>.

Process modularity over time: modelling process execution as an evolving activity network

*Pedro Parraguez

DTU Management Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark
(e-mail: pedro@dataverz.net).

Sebastiano A. Piccolo

DTU Management Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark
(e-mail: sebpi@dtu.dk).

Marija Majda Perišić

Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, 10000 Zagreb, Croatia
(e-mail: marija.perisic@fsb.hr).

Mario Štorga

Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, 10000 Zagreb, Croatia, and also with the Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, 97187 Luleå, Sweden (e-mail: mario.storga@fsb.hr).

Anja M. Maier

DTU Management Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark
(e-mail: amai@dtu.dk).

Abstract

Process modularity describes the extent to which processes can be decomposed into modules to be executed in parallel. So far, research has approached process modularity from a static perspective, not accounting for its temporal evolution. As a result, the understanding of process modularity has been limited to inferences drawn from aggregated analyses that disregard process execution. This paper introduces and develops the notion of dynamic process modularity considering the evolving activity network structure as executed by people. Drawing on network science, the paper quantifies process modularity over time using archival data from an engineering design process of a biomass power plant. This paper shows how studying the temporal evolution of process modularity enables a more complete understanding of activity networks, facilitates the comparison of actual process modularity patterns against formal engineering design stages, and provides data-driven decision-support for process planning and interventions. Finally, managerial recommendations for interface management, resource allocation, and process decomposition are proposed, to help practitioners better to understand and manage dynamic processes.

Keywords: modularity; process modularity; network analysis; design process

1. INTRODUCTION

Modularity as a design principle and as a property of complex systems has been used to manage the design process of rapidly evolving technologies, to achieve faster development cycles, and to better address customer demands [1], [2]. Modularity has become increasingly important because of the ever-greater complexity of modern engineering systems [3]–[6] and has consequently been formulated, adopted, applied, and discussed in multiple research fields including engineering (for reviews see [7], [8]) and management (for a review see [9]). Modularity is also often described as a design strategy for building and managing complex systems effectively [10] and it has been associated with desirable product or system properties and with organisational performance [11]. For example, in engineering, the concept of modularity is used for designing products, organisations and processes, as well as to obtain advantages such as maintainability, parallelisability, and manageability [12]–[19].

However, despite the frequent use of modularity in the fields of engineering and management, two aspects have remained under-researched. One, as noted by [1], [9], [20], so far, most efforts have been focused on product and organisational modularity, with comparatively fewer studies that systematically analyse the modularity of processes. Two, a broadly accepted quantitative measurement methodology for modularity is missing and the methods that are currently applied do not provide a satisfactory description of complex socio-technical systems and their architecture [21]. This is further complicated by an intrinsically dynamic nature of the process domain [22].

The objective of the work presented here is to develop a notion of process modularity that enhances the understanding of the process *as executed*, allowing comparisons with the process *as planned*. Coherent with the representation of the process as an activity network, this paper presents the *execution* of a process and its unfolding as a dynamic (time-evolving)

activity network.

In what follows, this paper provides empirical evidence that the modularity of the process, *as executed*, changes over time and exhibits temporal dependency. In addition, it shows the limitations of static analyses where the evolution of the process is not taken into account. Moreover, through a regression analysis, it is shown how the number of people, the number of tasks, and the number of interfaces between the tasks relate to process modularity. Finally, it is also shown that the number of modules alone does not provide sufficient information to capture the degree of modularity. These findings are used to derive managerial implications useful to better control process modularity.

The quantitative insights are complemented with a qualitative comparison of the real modularity profile and the theoretical expectations for the analysed case. As such, the paper highlights how the proposed approach is useful to compare the process as planned with the process as executed. Finally, the paper discusses the importance of findings and insights for engineering project managers looking for methods and indicators to contrast plans and execution.

The remainder of this paper is organised as follows. Section 2 provides a background on process modularity. Section 3 introduces process modularity, its expected dynamics through the process, and the methods to measure it. Analysing archival data, section 4 presents the results of quantifying modularity over time for the design process of a biomass power plant. The paper then compares the results of the dynamic analysis against the static version, where modularity is measured on a static time-aggregated network. Section 5 discusses and connects the findings to extant literature and previous findings, thus providing external validation. Finally, section 6 concludes the paper highlighting the relevance of a dynamic view of process modularity.

2. BACKGROUND ON MODULARITY

Systems can be studied through their network representation, focusing on the way elements are interconnected [23]–[25]. This perspective has shown that many real networks are *complex*; that is, their topology is rather different from random homogeneous networks [25]. This perspective has been applied to the study of engineering systems, generating a plethora of insights (e.g. [26]–[28]).

Engineering systems tend to exhibit the small-world property [29]–[31] and right skewed degree distributions [29], [30]. Small-world means that the average shortest path length, a measure of separation between any two nodes in the network [31], is small and the clustering coefficient, a measure of cliquishness [31], is high if compared to the clustering coefficient of a random network with the same number of elements and links. A right skewed degree distribution means that there is a small number of elements with many connections and a large number of elements with few connections. This heterogeneity in the degree distribution has important implications regarding the robustness of the network [28], [30], [32] and can be leveraged to devise intervention strategies, targeting the nodes with higher degree, to increase the resilience of the system [28], [30], [33]. Additionally, complex networks are typically sparse as their edge density is small [25], [26], [30], disassortative as high degree nodes tend to connect to low degree nodes [26], [34], and decentralised to avoid super-hubs and foster a higher functional decomposition [30], [35], [36]. These three characteristics are associated with higher resistance to error propagation and failure tolerance [28], [30], [36].

It is useful to note that generating networks that minimise the edge density, the average path length, and that increase the level of decentralisation produces modular networks. This supports the view that many engineering systems are the result of deliberate design and optimisation [35]–[37]. In general terms, modularity indicates the degree of decomposability

of a given system, dependent on the level of analysis and defined system granularity [38]–[40].

A system is *modular* if it is divisible into groups of components where intra-group connections are stronger than inter-group connections and where each function of the system typically maps on to one module [41]. In contrast, a system is *integrative* if it is not easily divisible into such modules and has inter-group connections which are almost as strong as the intra-group connections [41].

These definitions show that the problem of discerning modular and integrative systems is challenging in two ways. Firstly, a method to divide a system into partitions is needed and secondly, a method to measure and compare the strength of intra-partition links and the strength of inter-partition links is also needed [42], [43].

Many algorithms exist to identify partitions in a network, sometimes referred to as clusters, modules or communities. Network partitioning is a large and continuously evolving research topic within network science (for comprehensive reviews see [44], [45]). Similarly, in the engineering design community, several partitioning methods have been developed and utilised through applications of the Design Structure Matrix (DSM) (e.g. [46], [47]). Once the network partitions have been defined, comparison of the strength of intra and inter-partition links are possible, to measure the degree of modularity. Depending on the conceptualisation and definition of modularity, different mathematical formalizations are possible [44].

However, simultaneously calculating the number of partitions and the strength of intra and inter-partition links between the partitions requires a way to solve the circularity between these two measurements. For instance, comparing the simple count of inter-partition links against the simple count of intra-partition links would not be satisfactory. Indeed, this simple comparison would have the side effect that its theoretical maximum would be reached for a single cluster that contains the whole system, as there would be no inter-partition links; therefore, similar hypothetical metrics would fail to describe the modularity of the system.

One solution to this problem is offered by the field of network science, with the use of network null models as means to compare observed and expected metrics in a network. Specifically, modularity has been defined as the difference between the observed number of intra-module connections and the same quantity if connections were assigned at random while preserving the number of connections for each node [42], [43]. If the observed and expected number of intra-module connections are the same, a network has no evidence of a modular structure. Such a measure of modularity has allowed the development of algorithms to find partitions in networks by direct optimisation of the measure itself [42], [43], [48].

2.1 Modularity approaches applied to the process domain

Processes are often represented as activity (or task) networks of directed graphs where links between nodes represent technical dependencies or expected information flows. In the context of processes, modularity has been studied through activity networks with the purpose of modularising the process, by identifying groups of activities and by optimising their sequence [47], [49]. In turn, the quantification of the degree of modularity has relied on a combination of standard and ad-hoc network metrics to measure the degree of local modularity, e.g. [50] use of component modularity and of global modularity, [51] use of the modularisation function, and [29] use of the graph clustering coefficient. Such applications of modularity in the process domain have helped to increase process flexibility through activity re-sequencing, to reduce costs through standardisation, to identify the existence of sub-processes, to define which sub-processes can be run in parallel, and therefore to support overall process planning work [46], [52], [53]. In addition, process modularity is sometimes discussed in the context of process changeability, flexibility, and reliability studies where modularity appears as one of the studied aspects, e.g. [54]–[57].

To-date, process modularity has been applied mainly for planning purposes and, as such, applications of process modularity to monitor the dynamics of process execution are still missing. As the execution of the process often departs from the original plan due to changes, iterations [58], [59], the way people are assigned to activities [30], local and system task decomposition [60], or stage overlapping [46], the temporal investigation of a metric such as process modularity becomes a useful indicator to monitor the process. This leads to a better understanding of periods where the execution of the process diverges from the planned process. In addition, understanding the modularity of process execution can indicate periods in which concurrency can be increased as the process network is more modular in comparison to other periods. Indeed, [61] and [56] have shown that concurrency has limits and that due to linkages between modules, a higher number of concurrent process modules does not necessarily translate into a shorter execution time. As such, an approach to understand and track modularity during process execution using a formula that accounts for patterns of connectivity in the network is important and leads to detection of areas of intervention.

2.2. Process modularity from a dynamic and socio-technical process perspective

From a technical perspective, the architecture of the design process can be represented by a network of information dependencies between design tasks; a network that helps to consider logical precedencies and to define the sequence of activities based on information needs [62]. This technical focus is the one practised by most studies of modularity in the process domain. Such studies have an emphasis on process planning, derive technical and information dependencies primarily from the process architecture, and use a static network model to study clusters, to partition the network, and to assess overall modularity.

In contrast, from a socio-technical perspective, the architecture of the design process

can be understood as a network of interdependent activities executed by people over time. This perspective sees processes as "a series of actions or steps taken in order to achieve a particular end" [63]. Within engineering design, the design process is described as "the network of activities performed with the goal of producing a design [...] A design process is a real, actual way in which design work is done and designs are produced" [64]. Although such a socio-technical model of the design process has been previously proposed and operationalised through network analysis [30], [58], [65]–[67] the quantitative analysis of process modularity over time using such a model has not been implemented yet.

Dynamic socio-technical models of the design process (e.g. [58], [65]) also provide a more comprehensive view of design processes, allowing to quantify the evolution of process modularity and opening opportunities to investigate whether:

- The modularity of engineering processes changes through the different system engineering stages in response to different technical and/or organisational requirements associated with each stage.
- Changes in the network properties described in section 2 relate to changes in process modularity.
- The modularity of the actual process is connected to the modularity of the planned process, the product and the organisation (and in what way).
- Changes over time in the number of modules and degree of modularity provide a relevant description about the current state of a process.

2.3. The dynamics of process modularity in complex engineering design processes

Generic system engineering models such as the INCOSE's Systems Engineering V-model (SE-V) [68], [69] and the design process stages described by [4] have been widely

adopted models of complex engineering processes.

Starting from the four generic stages described by the two process models above (i.e. conceptual design, system-level design, detailed design and system integration stages) and supported by the analysis of information flow patterns in [65], explicit expectations about the relative number of process modules and the relative degree of process modularity are described for the four generic system engineering stages. In the discussion section, such expectations derived from theory on the number of modules and degree of modularity are then used as a reference against which to evaluate the findings obtained from the empirical dynamic analyses of process modularity in this paper.

The conceptual design stage: At this stage, the feasibility of the product is addressed, a design concept is developed, and early functional or proof of concept prototypes might be built and tested. Here, in the early stage of the development process, the number of people and activities is likely to be small but increasing as the process gains complexity. Due to the need for an overall and integrative design view, low levels of hierarchical decomposition at the process level are expected. As a result, both the number of process modules and the degree of process modularity are expected to be low.

The system-level design stage: During this stage, the design is refined, and major sub-systems and interfaces are defined leading to slightly higher levels of specialisation. Here, the number of people and activities is likely to increase as more specialised functions are defined. Due to the increase in the number of activities and people involved, as well as the increase in technical specialisation, hierarchical decomposition is expected to increase. As a result, the number of process modules and the degree of process modularity are expected to increase.

The detailed design stage: At this stage, the focus shifts to each single sub-system, defining the finest details for each designed component. Here, the number of people and activities involved in the process is likely to reach its peak in response to the need for

concurrent work in multiple and highly specialised design areas. Such a high level of specialisation leads to reaching the maximum hierarchical decomposition. As a result, the number of process modules and the degree of process modularity is expected to reach a peak.

The system integration stage: During this stage, most testing and integration activities are performed. Here, the number of people and activities starts to decrease. The high levels of technical specialisation of the previous stage are replaced by a more integral understanding of the entire system and the interactions between the different parts, which decreases the hierarchical decomposition of the process. As a result, the number of process modules and the degree of process modularity is expected to decrease.

3. Data and Methods

This section presents the data (3.1), the methods used to model the time evolving architecture of an engineering process by means of a temporal network (3.2), and the algorithm and equation to quantify the temporal evolution of process modularity (3.3).

3.1. Data

This paper uses data from the engineering design process of a biomass power plant. The design process is documented in [65], [70]. The dataset is a log that registers each time a person submits a new document or edits a previously existing document within the company's document archival system. As each document has an explicit link with a design activity [71], it is possible to derive a dynamic network that describes the actual process execution, mapping people and activities over time. The examined dataset includes the entire design process of one renewable energy plant from beginning to end. The studied process includes activities in areas such as the design of the boiler, combustion system, hydraulic and cooling system, pipes and

pressure parts, air and flue gas, steel structures as well as integrative work activities such as overall project management and on-site coordination. The records amount to more than 3000 documents logged in the system and include more than 130 unique activity codes and 80 people in total during the process.

Since using a high level of detail is especially important in the analysis of modularity [38], [72] activity codes containing the highest level of detail currently applied by the company for the purposes of project planning, management, budgeting, and control were collected. Examples of activity codes include “definition of process flow diagrams (PFD)” and the “design of boiler support structures”.

Person-activity relations were extracted from the dataset and for each relation, the time-range in which the person performed an activity when working on a specific task was computed. This method is inspired by analyses of event logs applied in the discipline of process mining [73], [74]. As the dataset does not include a detailed account of the time that each person spends working on the assigned activities, a ‘person-activity’ link is considered active between the first and the last time the link appears. In this case, as the dataset registers document metadata, the first and the last appearances of a person-activity link correspond to the creation date and the last modification date of some documents. Considering the dynamics and properties of this process highlighted and discussed in previous work [30], [58], [65], [70], [75] this simplification appears appropriate.

Interpretations of the development process and the process architecture derived from this analysis were further informed and validated in three ways: a) through 10 retrospective interviews with project managers and engineers, including the Vice President of Operations, the Vice President of Engineering, the technical project manager, the QA/QC manager, a QA/QC engineer, the site manager, the procurement manager; b) through the revision of formal project documentation including Gantt charts and workflow diagrams, which is used as a

benchmark to test the alignment of the planned process modules and the actual process modules; and c) through previous analyses of this case development process, including characterising process interfaces [70], information flows [65], process robustness [30], and iterations and stage-transitions [58].

3.2. Modelling the time-evolving architecture of the process

Following Sim and Duffy's generic ontology of design activities [71], engineering processes are conceptualised as a set of interconnected activities performed by people, whose purpose is the definition and evaluation of engineering system components, or the coordination and management of the process itself.

The evolution of the design process is represented using a temporal network of people and activities extracted from daily records of the activities that each person was working on. Such a temporal network can be thought of as a sequence of networks or, equivalently, as a sequence of Design Structure Matrices (DSMs) that represent each time period. To analyse the network snapshots the paper follows a method described in [76], [77] where the properties of interest for each snapshot are extracted to study changes over time.

In this case, each snapshot is a bipartite network of people and activities [30]. Therefore, to allow an interpretation consistent with process DSM [78], which considers the way activities are interconnected, the bipartite network is projected onto an activity-activity network. With such a projection, the coupling between activities is given by the number of people that activities have in common. In this model activities that are performed in the same time-period by the same group of people are more likely to be from a functional- and organisational-perspective part of the same module, since they draw directly from the same pool of knowledge and know-how.

The modularity value for each network snapshot is computed and the time series is plotted, along with the respective number of modules, number of people, and number of activities for visual comparison. In the next section, the methods used to compute modularity of each snapshot are described.

3.3. Quantifying process modularity

Using the one-mode network and a community detection algorithm is possible to detect modular structures and measure their degree of modularity. This paper calculates the degree of modularity using a measure known as *modularity* [42], [43] which measures the quality of a network partition in terms of modules separability. The modularity of a partition is a scalar value between -1 and 1 that measures the density of links inside modules compared to their expected quantities; this measure of modularity for a weighted graph (indicated with Q), is defined as follows:

$$Q = \frac{1}{2w} \sum_i \sum_j (w_{ij} - \frac{w_i w_j}{2w}) \delta(C_i, C_j) \quad (1)$$

Where $\delta(C_i, C_j)$ is the Kronecker function, which takes the value of 1 if the activities i and j are within the same module or 0 if they are not; w_{ij} is the weight on the edge between activities i and j ; w_i and w_j are their strengths [79]; C_i and C_j are the modules to which activities i and j are assigned, and w is the total strength.

Formally, the strength of a node i is defined as:

$$w_i = \sum_{j \neq i} w_{ij} \quad (2)$$

The total strength is, instead, the sum of the nodes' strengths:

$$w = \sum_i w_i \quad (3)$$

The heuristic chosen to reveal the modular structure of the process is the multilevel algorithm [48] also known as Louvain method, which is a direct optimisation of the modularity function. This combination is chosen as the formula of modularity (Q) captures the idea that the more separable the modules, the higher the network modularity, while the Louvain algorithm was selected as it has been shown to be able to find partitions of good quality [44], [48], [80]. Additionally, section 4.1 shows that the Louvain method is appropriate for the examined data and captures real modules with a high degree of accuracy.

The application of Louvain method and modularity is exemplified in figure 1 in both matrix and network diagram forms, showing a subset of a network of tasks previously reported in [66]. The eight tasks are labelled from A to H. The Louvain method correctly identifies two process modules and a modularity of 0.22. If the edges “E-F” and “D-H” are removed from the network and modularity is re-calculated, the value of modularity is now 0.34 (an increase of more than 50% over the previous network). This small example shows that the modularity equation adopted in this paper captures the idea that modularity should be higher if there are fewer connections between the modules.

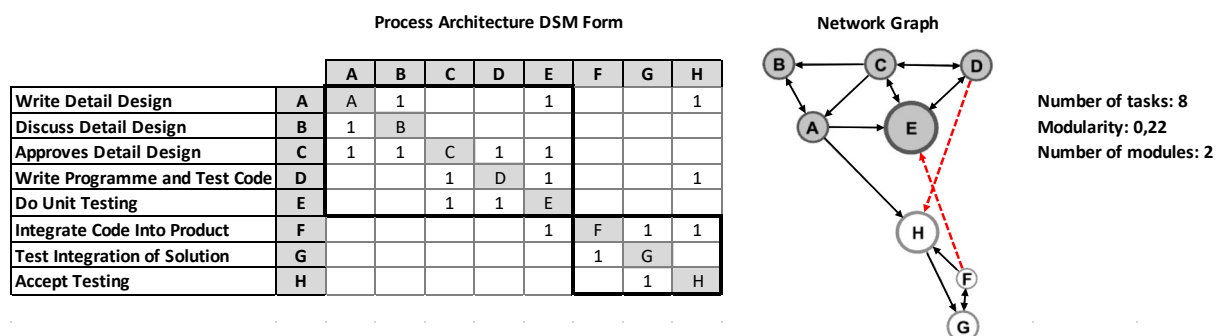


Figure 1: On the left, a Design Structure Matrix (DSM) of a network segment illustrating "Design and Integration" iterative activities of a software development process studied in [66] (inputs in columns). The two boxes outlined with a thicker line in the matrix show the two network modules identified. On the right, a graph representation of the same network. Each node represents an activity and is sized according to in-degree. The two modules are differentiated by white and grey circle shading. The two red-dashed lines highlight the edges

that were removed in the second part of the analysis.

4. Results

This section shows the value of analysing process modularity over time to track and understand process execution. It starts by performing a baseline analysis of a time-aggregated network where the network evolution is ignored (section 4.1), to later compare it against the additional information produced by the temporal analysis. To show that the measurement of modularity over time facilitates the comparison to process plans or models of design processes, section 4.2 contrasts the obtained profile with the expected modularity dynamics derived from theory, discussed in section 2.3. Finally, a regression analysis is performed to understand the relations between modularity and network properties (section 4.3).

4.1. Process modularity of the time-aggregated network and the need of a temporal analysis

The application of the Louvain method in the static network, where all the historical information is aggregated into one network discarding its temporality, identified six modules and an overall degree of modularity of 0.27 for the entire process. This value indicates low separability between the modules, despite the significant modular structure of the network. Indeed, the graphical representation in figure 2A shows that the amount of between modules linkages, in grey, is in relative terms high. A closer inspection of the modules shows that activities related to project management and quality control are distributed across all modules, indicating that those activities operate both at the system and sub-system levels, which is in accordance with the results reported in [60].

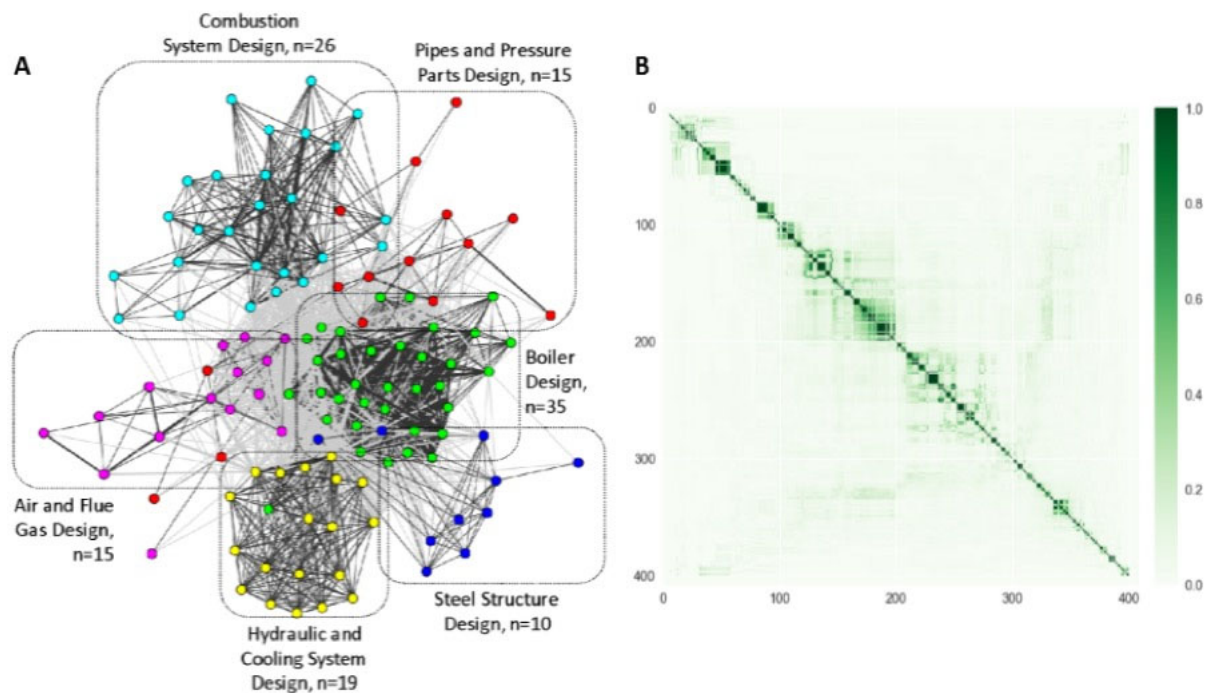


Figure 2: A) Network diagram showing the modules identified by the Louvain method on the time-aggregated network. The identified activity modules, labelled in accordance with the predominant activity type in each module, correspond to major design areas and sub-systems. B) Heatmap of similarity between the daily snapshots of the process network: the darker the more similar two snapshots are. The heatmap shows that the structure of the network changes significantly over time and the similarity is higher only for small periods.

To quantify the goodness of the partitions, the authors evaluated the overlap between the modules identified and the official modules defined by the company in their description of the process and sub-processes (ground truth). For this, the Rand index (R) [81], the adjusted Rand index (ARI) [82], and normalised mutual information (NMI) [83] were used. As a result, it was found that there is high agreement between the modular structure found by the Louvain method and the ground truth: $R = 0.93$, $ARI = 0.75$, $NMI = 0.82$ (values of 0 indicate that the two sets do not agree on any pair of points and values of 1 indicate that the sets are the same). As such, the Louvain method is able to recover a meaningful modular structure that is consistent with internal company descriptions of their design process, confirming the appropriateness of this method for this study.

To show that the analysis of the time-aggregated (static) network is not informative of the dynamics of the design process and does not allow understanding of how to design and

manage effective process networks, the authors analysed how the network topology changes over time in the temporal network. To perform this analysis and obtain the results shown in figure 2B, the topological overlap was calculated as described in [76], [77]. First, it can be noted that the average similarity between each snapshot and the time-aggregated network is 0.035 ± 0.002 . This means that on average for each daily snapshot, less than 4% of edges are present. Second, figure 2B shows that the similarity of the snapshots over time is low, except for some short periods (the average similarity between two snapshots is 0.046 ± 0.002).

This indicates that the temporal network has rich dynamics and that the time-aggregated network is unsuitable to study them. As the network topology changes significantly over time, there are reasons to believe that process modularity would change as well. Therefore, the next two sections investigate the dynamics of process modularity over time both qualitatively and quantitatively.

4.2. Process modularity results for the evolving network

Here, process modularity is computed for each temporal snapshot, using daily resolution, to obtain a modularity profile over time. The modularity profile is also compared with the evolving number of people, number of activities, and number of process modules. In addition, a qualitative comparison with the theoretical expectations for modularity and number of modules for each design process stage is offered (section 2.3).

Figure 3 shows the temporal profile of modularity, number of people, number of activities and number of modules as well as the project stages as defined by the company, including both the data points and the trend line for each period. The white space between conceptual design and system-level design (between April 2010 and October 2011) shows a time when the process was put on hold due to negotiations and search for investors. Hardly any document was created or completed during that period.

Although the first two stages contain few data points, especially in comparison with the other two stages, they provide a global picture of the process evolution at that point in time.

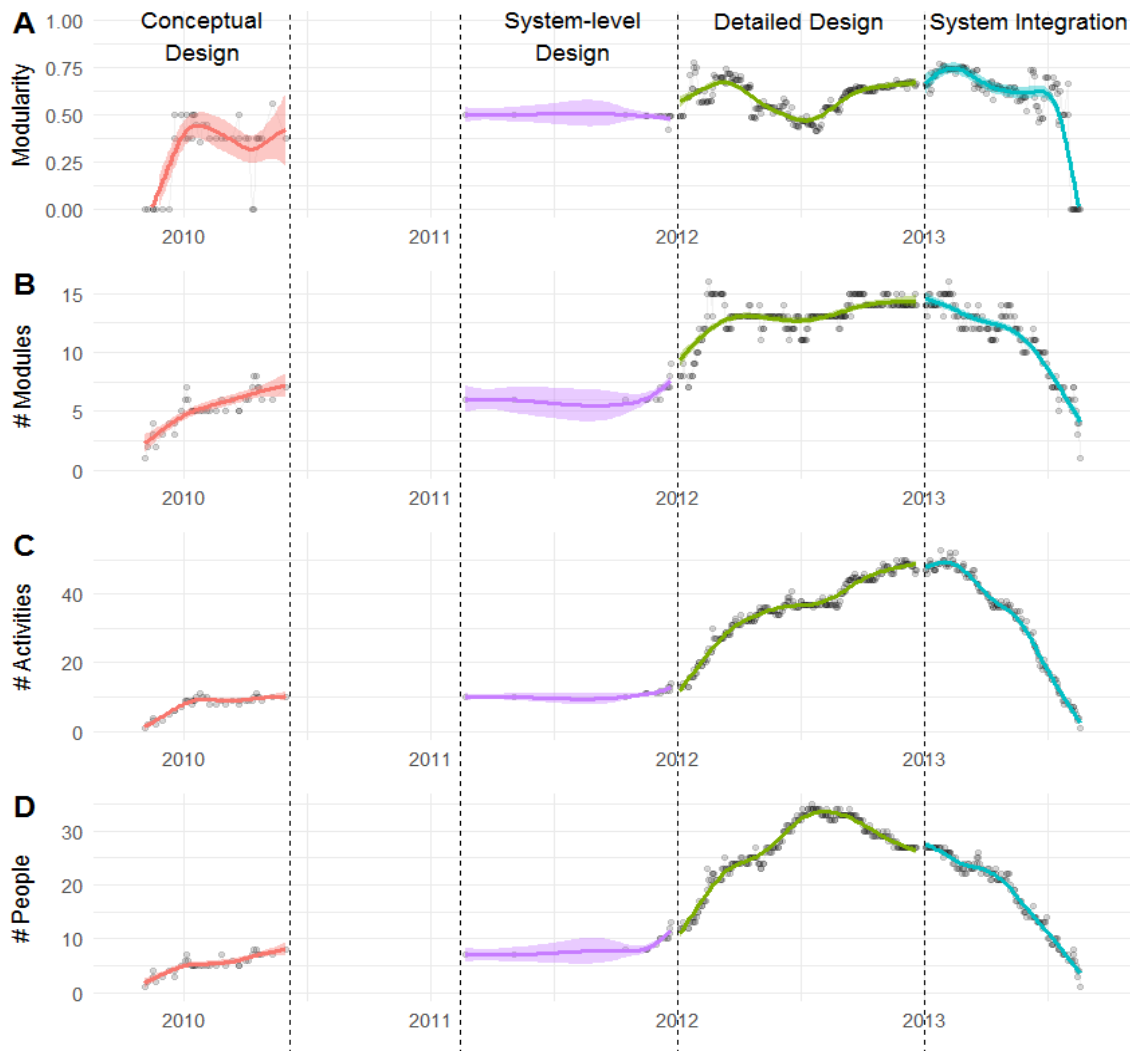


Figure 3: Temporal profiles for process modularity (A), the number of modules (B), the number of activities (C) and the number of people (D) involved in the process over time.

The stage of conceptual design is characterised by a relatively low number of people working on a low number of activities. The number of modules is high if compared with the number of activities in the process, indicating the presence of isolated activities. Process modularity lies in the range $[0, 0.56]$ with most of the points in $[0.37, 0.5]$. Consistent with the theoretical expectations, this portrays a stage where activities form a small and cohesive

network. From previous work [84], it is known that in this stage, the focus was mainly on boiler design, 3D models of the power plant and negotiations of functional requirements and specifications.

During system-level design, the number of people, number of activities, and number of modules experience a small increase, but only towards the end of the stage. This is due to the preceding period where the process was on hold in order to find new investors. Indeed, in previous work [84], it was found that after finding the new investors, the company re-planned the project to accommodate changes on the boiler requested by the client and worked on functional specifications and calculations in order to be ready for the detailed design stage. The marginal increase in the degree of modularity, compared to the preceding stage, can be interpreted as a sign of progressive hierarchical decomposition consistent with the need of breaking down the overall system into its major components in order to tackle the detailed design.

In the detailed design stage, the number of people and number of activities increase considerably, and the number of people reaches its maximum value. The number of modules oscillates between 10 and 16. Process modularity lies in the range [0.42, 0.78] and, at the beginning of the stage, experiences a sudden change of level. In [84] it was found that the focus of the work shifted toward the components of the power plant such as membranes, grates, air preheater, flue gas condenser, and valves.

A strong decrease in process modularity appears in the period between May and August 2012, indicating more integration between the activities. In this period, the company began the development of quality control plans, experienced missing deliveries, and problems with the welding procedures of the piping system that generated changes and iterations. These events and the increased iteration during this period are in line with a higher degree of integration between the process modules and thus the observed decrease in modularity. The successive

increase in both the number of modules and modularity is consistent with the increase in functional specialisation and more separability of process modules expected for this stage.

During system integration, the number of people steadily decreases. The number of modules and the number of activities reach their maximum and then decrease steadily. Process modularity increases until it reaches a maximum and then decreases again. This pattern is a sign of a progressive merging of modules and therefore of functional integration, which is consistent with the expectations for this stage of the process. This is also confirmed in previous work [84] which found that during this stage the work was oriented toward an important milestone of the project: the ‘main pressure test’, which is a comprehensive test of all the main connection pipes, boiler, drum, super-heaters, vents, drains, and other components. The integrative nature of the work needed for this test is consistent with the observed decrease of modularity.

4.3. Statistical analysis of process modularity dynamics

The previous section showed insights provided by the analysis of process modularity over time. This section seeks to understand which network characteristics are related to modularity in order to derive managerial implications. Here, the qualitative analysis is complemented with a regression analysis, using heteroscedasticity robust standard errors, to model process modularity as a function of the number of people, number of activities, number of modules, network density, average path length, clustering coefficient, centralisation, and assortativity. Time dependency is controlled by including the value of process modularity at the previous time point in the model.

Table 1: Regression results

	Model 1	Model 2	Model 3	Model 4
Intercept	0.62 *** (0.00)	0.62 *** (0.00)	0.62 *** (0.00)	0.62 *** (0.00)
Modularity _{t-1}	0.08 *** (0.00)	0.05 *** (0.01)	0.04 *** (0.00)	
Modules		-0.01 *** (0.00)		
People		-0.01 ** (0.00)	-0.02 *** (0.00)	-0.04 *** (0.00)
Activities		0.03 *** (0.00)	0.16 *** (0.03)	0.27 *** (0.02)
Density		-0.02 *** (0.01)		
Clustering		0.01 (0.01)	0.03 (0.03)	
AvgPath		-0.01 *** (0.00)		
Assortativity		-0.01 (0.01)		
Centralization		-0.02 *** (0.00)		
log(Modules)			-0.03 *** (0.00)	-0.03 *** (0.01)
Activities ²			-0.11 *** (0.02)	-0.17 *** (0.02)
exp(Density)			-0.03 *** (0.00)	-0.03 *** (0.00)
Clustering ²			-0.02 (0.03)	
log(AvgPath)			-0.01 *** (0.00)	-0.02 *** (0.00)
exp(Assortativity)			-0.02 *** (0.00)	-0.02 *** (0.00)
log(Centralization)			-0.02 *** (0.00)	-0.05 *** (0.00)
N	334	334	334	334
R ²	0.86	0.92	0.94	0.89
AIC	-1342.26	-1501.72	-1590.57	-1405.26
BIC	-1330.82	-1459.79	-1541.03	-1367.12

Variables are standardised. Standard errors in parentheses are heteroscedasticity robust. *** p < 0.001; ** p < 0.01; * p < 0.05.

Table 1 highlights the results of the regression analysis. Model 1 shows the time dependency of process modularity: everything else equal, a higher process modularity at time

$t-1$ would result in a higher process modularity at time t . This simple model has already high predictive power.

Model 2 is a baseline where all the variables are in linear form. The number of modules shows a negative association with process modularity, meaning that in the process here analysed when the number of modules increases, the degree of separability between modules decreases. The number of people negatively correlates with process modularity. Everything else equal, increasing the number of people working on the tasks is expected to decrease process modularity. This happens because, in the examined process, people are allocated to multiple activities; therefore, increasing the number of people increases the number of links between the activities, which in turn decreases the separability between the modules. The number of activities in the network is positively associated with process modularity; conversely, the density of links is negatively associated with process modularity.

Everything else equal, an increase in the number of tasks or a decrease in the number of edges decreases the density, which is expected to produce an increase in process modularity. The average path length is negatively associated with process modularity. This means that in the process, modularity does not penalise the efficiency of the network to transfer information. Process modularity is also negatively correlated with network centralisation. That is, when the process becomes more modular it becomes also more decentralised. Finally, in this model, clustering coefficient and assortativity are not statistically significant.

Model 3 accounts for non-linear relations between the network variables and process modularity. This model investigates the possible presence of turning points by adding quadratic terms and modelled non-linearity using exponential or logarithmic transformations. The model confirms the results from the second model, while assortativity becomes significantly associated, in a negative way, with process modularity. In fact, as assortativity increases,

activities with high number of interfaces tend to be connected together; therefore, the network becomes more cohesive. The effects of model 3 coefficients are plotted in figure 4.

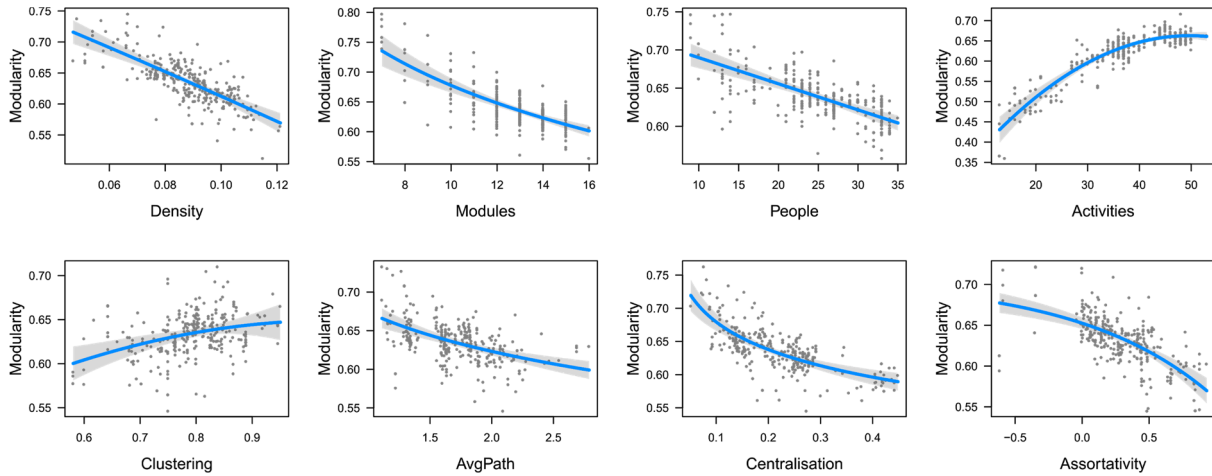


Figure 4: Conditional plots of the third regression model. Each subplot shows the association between the specific network property and modularity in the examined data. Besides the trendline (bold), the confidence bands are coloured in grey. The plots also show that the model fits the data appropriately.

Model 4 removes the clustering coefficient, which was not significant, and the lagged process modularity term to show that most of the explained variance is given by the network characteristics.

From the statistical analysis presented above the following findings can be derived. The number of people provides insights about resource allocation: increasing the number of people working on a process without changing the process structure or re-planning the process, is likely to be accompanied by the decrease in process modularity.

The number of activities gives insights about the level of specialisation and functional decomposition: a higher number of activities in the process network often corresponds with a higher degree of decomposition. Splitting a large task into two and re-organising the process accordingly can help to reduce the coupling between tasks and modules, thus increasing modularity.

In the examined process, modularity is associated with sparse, decentralised, efficient, and more resilient networks. This set of findings provides empirical evidence that modularity

optimises multiple constraints. Indeed, [37] shows that the generation of networks that increase robustness and that minimise both the total number of edges and the average path length, yields modular networks. These relations are also supported by the case data analysed here.

5. DISCUSSION

This section examines, discusses, and connects the results with extant literature, providing pointers for future research. It discusses the importance of a temporal analysis of process modularity, pointing to the limitations of aggregating temporal information into one static network (5.1). It compares similarity and differences between the observed temporal dynamics of process modularity and the theoretical expectations, showing that a dynamic analysis of process modularity is a useful tool to monitor design processes (5.2). Finally, connecting the results from the regression models with previous findings, this section derives implications for practice that are useful to manage process modularity (5.3).

5.1 Comparison of aggregated vs dynamic network analysis

The results presented here highlight the importance of analysing temporal dynamics of process modularity considering process execution. At the same time, the results also show the limitations of aggregating all the temporal information into a single static network.

The analysis of the aggregated network (figures 2B and 3) shows that process modularity and network structure significantly change throughout the design process. Furthermore, the average degree of similarity between a daily snapshot and the time-aggregated network is only 0.035 ± 0.002 . These considerations highlight the limitations of the time-aggregated network analysis, which is unable to provide insights about the execution of the

process and fails to describe the finer details of process modularity. This is consistent with previous studies in the context of social network evolution [76], [77].

In contrast to time-aggregated analysis, the dynamic temporal analysis reveals how process modularity changes over time, showing different patterns through the stages of the process under analysis. The reason for these changes is that certain connections between tasks are bounded in time and appear only in some periods following the progression of the process. Thus, the process modularity profile over time is useful not only to compare process execution with theoretical expectations or project plans (section 2.3) but also to obtain a data-driven view of process execution, providing modularity insights during process execution

5.2 Comparison between observed and expected process modularity over time

Despite the small number of data points for the first two stages, overall, the observed process modularity profile matches the expected modularity patterns described in section 2.3. For example, process modularity increases during the detailed design and decreases during the system integration stage.

In terms of departures from theoretical expectations based on formal process stages, two visible deviations from expectations can be observed: a decrease in modularity around half-way through the detailed design stage, which is due to unexpected iterations and project changes, and an increase in modularity that continues for almost three months after the formal beginning of the system integration stage. This suggests that iterations delayed the project and thus the actual beginning of system integration. Indeed, from previous work [84] it is known that the delay produced by those iterations required major efforts in coordination and many hours of overtime.

The previous example shows that comparing the observed process modularity profile

with the theoretical expectations can help to identify problems or departures from project plans and formal process stages including iterations or periods with coordination overhead. In addition, the profile that was obtained and the departure from the expectations previously discussed suggest the view of the design process as partially planned and partially emergent [6] integrating both iterative [59], [85] and stage-based views e.g. [4], [68], [69].

As the formula used to quantify process modularity evaluates the degree of separability of network modules, periods with higher modularity might be more suitable than other periods to increase process concurrency. In the examined case, such a period seems to be the detailed design stage. Future research may investigate whether this is the case and what limits might exist. In fact, increasing process concurrency requires a re-planning, which comes with costs and might also have negative effects [61].

5.3 Managerial implications and insights

The regression models show that as the process becomes more modular, its network architecture becomes sparser (lower density), more decentralised (lower centralisation), more efficient in the information transmission (smaller average shortest path length), with higher functional specialisation (higher number of activities and lower centralisation) and higher resistance to error propagation and potential for concurrency (lower assortativity). The models also show that increasing the number of people working on the process does not necessarily increase modularity. This result is consistent with the fact that adding people to a late project is likely to delay the project even further [86]. Thus, this paper adds to previous studies that suggest avoiding breaking functions in a process among too many people [61], [87]. This finding also points to the importance of central people for process robustness [30] suggesting to avoid increasing the number of people that manage interfaces between modules and to select

those people carefully as they can either facilitate or block the propagation of errors [30], [88].

Finally, the models show that counting the number of modules is not a measure of modularity and, in the examined data, when the number of modules increases modularity decreases because the modules are less separable. As such, it also corroborates insights from [61] that, in the context of concurrent engineering, have shown that a higher number of modules does not result in a shorter execution time as the linkages between modules would produce higher costs of communication and coordination.

Our results demonstrate that process modularity evolves over time during process execution and is affected by iterations, changes, random and adverse events. Therefore, this paper concurs with [89] in that the process should be made as adaptive and robust as possible to the perturbations that characterise its execution. This implies that the original plan should be routinely checked and adjusted accordingly. The results from the models allow us to derive some strategies to manage modularity during process execution. Modularity can be increased by making the network more decentralised and sparser; therefore, if the technical constraints allow, eliminating dependencies, especially between modules, is effective to increase modularity. This operation would be very similar to the operation of tearing [47], [78] and might require making assumptions or fixing the design of the interface before designing the modules. In line with this suggestion, sparsifying the product architecture by removing dependencies was previously found to be an effective strategy to increase product modularity [90]. The same effect can be obtained by increasing the level of functional specification by splitting an activity with many interfaces, also called hub, in two sub-activities. These hub activities have an important role in propagating errors [28], [30] and are associated with more iterations and completion time [58], [59]. Therefore, acting on these hubs is likely to bring multiple advantages beyond an increase in modularity. Hubs can also significantly change over time; therefore, it has been proposed that special attention should be given to hub-based

strategies [76], [77]. However, despite the high changeability of hubs, time-based strategies are not necessarily superior to hub-based strategies [91].

Additionally, the models show the importance of keeping the network efficient in terms of information transmission, avoiding increasing the average shortest path length and, when possible, to favour disassortative structures, avoiding to directly connect hubs together and using more hub-spoke topologies.

6. CONCLUSIONS

In this paper, process modularity over time was analysed to understand the process *as executed*. This perspective considers the assignment of people to activities. Connecting theoretical models for design stages, such as the Systems Engineering V-model (SE-V) [68], [69] and the design process stages proposed by [4], enabled a qualitative comparison between the observed temporal process modularity profile and its theoretical dynamics. In this case, it was found that departures from theoretical expectations could inform about the real progression of the project and highlight periods with problems, iterations, or changes. Thus, the temporal profile of process modularity facilitates the comparison between process execution and process plans. As a result, it has been shown that the temporal view of process modularity is important and that aggregating process modularity into a single static network only provides an overview about the overall way in which the process is organised, which is unable to describe the dynamic execution of the process.

The temporal profile of process modularity emphasised here was analysed to understand the relations between the number of people, activities, and interfaces between activities and process modularity. This allowed to derive suggestions for managers to manage process modularity showing that the number of people working on the process and the number

of interfaces between tasks negatively correlate with process modularity, while the number of tasks positively correlates with process modularity. Finally, a negative correlation between the number of modules and process modularity was found, showing that increasing the number of process modules without changing the structure of the activity network will not increase modularity, yet is likely to have an opposite effect.

References:

- [1] S. K. Vickery, X. Koufteros, C. Dröge, and R. Calantone, “Product Modularity, Process Modularity, and New Product Introduction Performance: Does Complexity Matter?,” *Prod. Oper. Manag.*, vol. 25, no. 4, pp. 751–770, 2016.
- [2] A. M. Ross and D. H. Rhodes, “Towards a prescriptive semantic basis for change-typeilities,” *Procedia Comput. Sci.*, vol. 44, no. C, pp. 443–453, 2015.
- [3] M. A. Schilling and H. K. Steensma, “The use of modular organizational forms: An industry-level analysis,” *Acad. Manag. J.*, vol. 44, no. 6, pp. 1149–1168, Dec. 2001.
- [4] K. T. Ulrich and S. D. Eppinger, *Product Design and Development*. New York, USA: McGraw-Hill Irwin, 2012.
- [5] A. Tripathy and S. D. Eppinger, “Organizing Global Product Development for Complex Engineered Systems,” *IEEE Trans. Eng. Manag.*, vol. 58, no. 3, pp. 510–529, Aug. 2011.
- [6] O. L. De Weck, D. Roos, C. L. Magee, and O. L. de Weck; Daniel Roos; Christopher L. Magee, *Engineering Systems: Meeting Human Needs in a Complex Technological World*, vol. 1. Cambridge: The MIT Press, 2011.
- [7] Baldwin and K. B. Clark, *Design Rules: The Power of Modularity*. Cambridge Massachusetts: MIT Press, 2000.
- [8] C.-C. Chen and N. Crilly, “From modularity to emergence: a primer on the design and science of complex systems,” 2016. [Online]. Available: <https://doi.org/10.17863/CAM.4503>.
- [9] D. Campagnolo and A. Camuffo, “The concept of modularity in management studies:

- A literature review,” *Int. J. Manag. Rev.*, vol. 12, pp. 259–283, 2010.
- [10] Baldwin and K. B. Clark, “Managing in an age of modularity,” *Harv. Bus. Rev.*, vol. 75, no. 5, pp. 84–93, 1997.
- [11] T. J. Marion and M. H. Meyer, “Organizing to achieve modular architecture across different products,” *IEEE Trans. Eng. Manag.*, vol. 65, no. 3, pp. 404–416, 2018.
- [12] K. Fujita, “Product variety optimization under modular architecture,” *Comput. Des.*, vol. 34, no. 12, pp. 953–965, 2002.
- [13] R. Rai and V. Allada, “Modular product family design: Agent-based Pareto-optimization and quality loss function-based post-optimal analysis,” *Int. J. Prod. Res.*, vol. 41, no. 17, pp. 4075–4098, 2003.
- [14] Baldwin and K. B. Clark, “Modularity in the Design of Complex Engineering Systems,” in *Complex Engineered Systems: Science Meets Technology*, D. Braha, A. A. Minai, and Y. Bar-Yam, Eds. Springer Berlin Heidelberg, 2006, pp. 175–205.
- [15] S. K. Ethiraj and D. Levinthal, “Modularity and Innovation in Complex Systems,” *Manage. Sci.*, vol. 50, no. 2, pp. 159–173, 2004.
- [16] A. Jose and M. Tollenaere, “Modular and platform methods for product family design: literature analysis,” *J. Intell. Manuf.*, vol. 16, no. 3, pp. 371–390, Jun. 2005.
- [17] H. Reijers and J. Mendling, “Modularity in Process Models: Review and Effects,” in *Business Process Management*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 20–35.
- [18] S. L. Pan, G. Pan, A. J. W. Chen, and M. H. Hsieh, “The dynamics of implementing and managing modularity of organizational routines during capability development: Insights from a process model,” *IEEE Trans. Eng. Manag.*, vol. 54, no. 4, pp. 800–813, 2007.
- [19] K. Park and G. E. Okudan Kremer, “An investigation on the network topology of an evolving product family structure and its robustness and complexity,” *Res. Eng. Des.*, vol. 30, no. 3, pp. 381–404, Jul. 2019.
- [20] M. Jacobs, C. Droge, S. K. Vickery, and R. Calantone, “Product and Process Modularity’s Effects on Manufacturing Agility and Firm Growth Performance,” *J.*

Prod. Innov. Manag., vol. 28, no. 1, pp. 123–137, Jan. 2011.

- [21] A. Cabigiosu and A. Camuffo, “Measuring Modularity: Engineering and Management Effects of Different Approaches,” *IEEE Trans. Eng. Manag.*, vol. 64, no. 1, pp. 103–114, 2017.
- [22] T. R. Browning, “Applying the design structure matrix to system decomposition and integration problems: a review and new directions,” *IEEE Trans. Eng. Manag.*, vol. 48, no. 3, pp. 292–306, 2001.
- [23] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge: Cambridge University Press, 2010.
- [24] A.-L. Barabási, “Emergence of Scaling in Random Networks,” *Science (80-.)*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.
- [25] A.-L. Barabási, “Statistical mechanics of complex networks,” *Rev. Mod. Phys.*, vol. 74, no. 1, p. 47, 2002.
- [26] D. Braha, “The Complexity of Design Networks: Structure and Dynamics,” in *Experimental Design Research*, Cham: Springer International Publishing, 2016, pp. 129–151.
- [27] D. Braha and Y. Bar-Yam, “Information flow structure in large-scale product development organizational networks,” *J. Inf. Technol.*, vol. 19, no. 4, pp. 244–253, Dec. 2004.
- [28] D. Braha and Y. Bar-Yam, “The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results,” *Manage. Sci.*, vol. 53, no. 7, pp. 1127–1145, Jul. 2007.
- [29] D. Braha and Y. Bar-Yam, “Topology of large-scale engineering problem-solving networks,” *Phys. Rev. E*, vol. 69, no. 1, p. 016113, Jan. 2004.
- [30] S. A. Piccolo, S. Lehmann, and A. Maier, “Design process robustness: a bipartite network analysis reveals the central importance of people,” *Des. Sci.*, vol. 4, p. e1, 2018.
- [31] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

- [32] R. Albert, H. Jeong, and A.-L. Barabási, “Error and attack tolerance of complex networks,” *Nature*, vol. 406, no. 6794, pp. 378–382, Jul. 2000.
- [33] A. Vázquez, R. Pastor-Satorras, and A. Vespignani, “Large-scale topological and dynamical properties of the Internet,” *Phys. Rev. E*, vol. 65, no. 6, p. 066130, Jun. 2002.
- [34] M. E. J. Newman, “Assortative Mixing in Networks,” *Phys. Rev. Lett.*, vol. 89, no. 20, pp. 1–4, 2002.
- [35] S. Valverde, R. F. Cancho, and R. V Solé, “Scale-free networks from optimal design,” *Europhys. Lett.*, vol. 60, no. 4, pp. 512–517, Nov. 2002.
- [36] J. C. Doyle *et al.*, “The ‘robust yet fragile’ nature of the Internet,” *Proc. Natl. Acad. Sci.*, vol. 102, no. 41, pp. 14497–14502, Oct. 2005.
- [37] R. K. Pan and S. Sinha, “Modular networks emerge from multiconstraint optimization,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 76, no. 4, pp. 1–4, 2007.
- [38] N. Chiriac, K. Hölttä-Otto, D. Lysy, and E. Suk Suh, “Level of Modularity and Different Levels of System Granularity,” *J. Mech. Des.*, vol. 133, no. 10, p. 101007, 2011.
- [39] H. A. Simon, *The Sciences of the Artificial*. Cambridge: The MIT Press, 1996.
- [40] H. A. Simon, “The architecture of complexity,” *Proc. Am. Philos. Soc.*, vol. 106, no. 6, pp. 467–482, 1962.
- [41] M. E. Sosa, S. D. Eppinger, and C. M. Rowles, “Identifying Modular and Integrative Systems and Their Impact on Design Team Interactions,” *J. Mech. Des.*, vol. 125, no. 2, p. 240, 2003.
- [42] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E*, vol. 69, no. 2, p. 026113, Feb. 2004.
- [43] M. E. J. Newman, “Modularity and community structure in networks,” *Proc. Natl. Acad. Sci.*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.
- [44] S. Fortunato, “Community detection in graphs,” *Phys. Rep.*, no. 486, pp. 75–174, Jun. 2010.

- [45] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *J. Netw. Comput. Appl.*, vol. 108, no. February, pp. 87–111, 2018.
- [46] A. Yassine and D. Braha, "Complex Concurrent Engineering and the Design Structure Matrix Method," *Concurr. Eng.*, vol. 11, no. 3, pp. 165–176, Sep. 2003.
- [47] S. D. Eppinger and T. R. Browning, *Design Structure Matrix Methods and Applications*. Cambridge: The MIT Press, 2012.
- [48] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 10, p. P10008, Oct. 2008.
- [49] H. Seol, C. Kim, C. Lee, and Y. Park, "Design Process Modularization: Concept and Algorithm," *Concurr. Eng.*, vol. 15, no. 2, pp. 175–186, 2007.
- [50] M. E. Sosa, S. D. Eppinger, and C. M. Rowles, "A Network Approach to Define Modularity of Components in Complex Products," *J. Mech. Des.*, vol. 129, no. 11, p. 1118, 2007.
- [51] J. H. Mikkola and O. Gassmann, "Managing modularity of product architectures: Toward an integrated theory," *IEEE Trans. Eng. Manag.*, vol. 50, no. 2, pp. 204–218, 2003.
- [52] R. I. van Hoek and H. A. M. Weken, "The Impact of Modular Production on the Dynamics of Supply Chains," *Int. J. Logist. Manag.*, vol. 9, no. 2, pp. 35–50, 1998.
- [53] J. Gualandris and M. Kalchschmidt, "Product and process modularity: improving flexibility and reducing supplier failure risk," *Int. J. Prod. Res.*, vol. 51, no. 19, pp. 5757–5770, Oct. 2013.
- [54] E. C. Y. Koh, N. H. M. Caldwell, and P. J. Clarkson, "A technique to assess the changeability of complex engineering systems," *J. Eng. Des.*, vol. 24, no. 7, pp. 477–498, 2013.
- [55] J. Hu and M. A. Cardin, "Generating flexibility in the design of engineering systems to enable better sustainability and lifecycle performance," *Res. Eng. Des.*, vol. 26, no. 2, pp. 121–143, 2015.

- [56] M. Salonen, K. H. Otto, and K. Otto, "Effecting product reliability and life cycle costs with early design phase product architecture decisions," *Int. J. Prod. Dev.*, vol. 5, no. 1/2, p. 109, 2008.
- [57] M. V Martin and K. Ishii, "Design for variety: developing standardized and modularized product platform architectures," *Res. Eng. Des.*, vol. 13, no. 2002, pp. 213–235, 2002.
- [58] S. A. Piccolo, A. M. Maier, S. Lehmann, and C. A. McMahon, "Iterations as the result of social and technical factors: empirical evidence from a large-scale design project," *Res. Eng. Des.*, Oct. 2018.
- [59] D. C. Wynn and C. M. Eckert, "Perspectives on iteration in design and development," *Res. Eng. Des.*, pp. 1–32, 2016.
- [60] A. A. Yassine, N. Joglekar, D. Braha, S. D. Eppinger, and D. E. Whitney, "Information hiding in product development: the design churn effect," *Res. Eng. Des.*, vol. 14, no. 3, pp. 145–161, Nov. 2003.
- [61] G. M. Hoedemaker, J. D. Blackburn, and L. N. Van Wassenhove, "Limits to Concurrency," *Decis. Sci.*, vol. 30, no. 1, pp. 1–18, Jan. 1999.
- [62] P. J. Clarkson and J. R. Hamilton, "'Signposting', A Parameter-driven Task-based Model of the Design Process," *Res. Eng. Des.*, vol. 12, pp. 18–38, 2000.
- [63] A. Stevenson, Ed., *Oxford Dictionary of English*. Oxford: Oxford University Press, 2010.
- [64] P. J. Clarkson and C. M. Eckert, *Design process improvement: A review of current practice*. London: Springer London, 2005.
- [65] P. Parraguez, S. D. S. D. Eppinger, and A. M. A. M. Maier, "Information Flow Through Stages of Complex Engineering Design Projects: A Dynamic Network Analysis Approach," *IEEE Trans. Eng. Manag.*, vol. 62, no. 4, pp. 604–617, Nov. 2015.
- [66] M. E. Sosa, "A structured approach to predicting and managing technical interactions in software development," *Res. Eng. Des.*, vol. 19, no. 1, pp. 47–70, 2008.
- [67] M. Kreimeyer and U. Lindemann, *Complexity Metrics in Engineering Design*:

Managing the Structure of Design Processes. London: Springer, 2011.

- [68] INCOSE, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, version 4.0*. 2015.
- [69] C. Haskins, K. Forsberg, M. Krueger, D. Walden, and R. D. Hamelin, *Systems Engineering Handbook*, no. October. San Diego, CA: International Council on Systems Engineering, 2011.
- [70] P. Parraguez, S. Eppinger, and A. Maier, “Characterizing Design Process Interfaces as Organization Networks: Insights for Engineering Systems Management,” *Syst. Eng.*, vol. 19, no. 2, pp. 158–173, Mar. 2016.
- [71] S. K. Sim and A. H. B. Duffy, “Towards an ontology of generic engineering design activities,” *Res. Eng. Des.*, vol. 14, no. 4, pp. 200–223, Nov. 2003.
- [72] S. Li, “A matrix-based clustering approach for the decomposition of design problems,” *Res. Eng. Des.*, vol. 22, no. 4, pp. 263–278, Apr. 2011.
- [73] W. M. P. Van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Berlin: Springer, 2011.
- [74] W. Van der Aalst, “Process Discovery: Capturing the Invisible,” *IEEE Comput. Intell. Mag.*, vol. 5, no. 1, pp. 28–41, Feb. 2010.
- [75] S. A. Piccolo, S. Lehmann, and A. Maier, “Using data- and network science to reveal iterations and phase-transitions in the design process,” in *Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 2: Design Processes | Design Organisation and Management*, 2017, pp. 11–21.
- [76] D. Braha and Y. Bar-Yam, “From centrality to temporary fame: Dynamic centrality in complex networks,” *Complexity*, vol. 12, no. 2, pp. 59–63, Nov. 2006.
- [77] D. Braha and Y. Bar-Yam, “Time-Dependent Complex Networks: Dynamic Centrality, Dynamic Motifs, and Cycles of Social Interactions,” in *Adaptive Networks, Understanding Complex Systems*, 2009, pp. 39–50.
- [78] D. V Steward, “The Design Structure System: A Method for Managing the Design of Complex Systems,” *IEEE Trans. Eng. Manag.*, vol. 28, no. 3, pp. 71–74, 1981.
- [79] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani, “The architecture of

- complex weighted networks.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 101, pp. 3747–3752, 2004.
- [80] S. Emmons, S. Kobourov, M. Gallant, and K. Börner, “Analysis of network clustering algorithms and cluster quality metrics at scale,” *PLoS One*, vol. 11, no. 7, pp. 1–18, 2016.
- [81] W. M. Rand, “Objective Criteria for the Evaluation of Clustering Methods,” *J. Am. Stat. Assoc.*, vol. 66, no. 336, p. 846, Dec. 1971.
- [82] L. Hubert and P. Arabie, “Comparing partitions,” *J. Classif.*, vol. 2, no. 1, pp. 193–218, 1985.
- [83] A. Strehl and J. Ghosh, “Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions,” *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2002.
- [84] S. A. Piccolo, J. Wilberg, U. Lindemann, A. Maier, and others, “Changes and Sentiment: a Longitudinal Email Analysis of a Large Scale Design Project,” in *DS92: Proceedings of the DESIGN 2018 15th International Design Conference*, 2018, pp. 869–880.
- [85] D. C. Wynn, N. H. Caldwell, and P. J. Clarkson, “Predicting Change Propagation in Complex Design Workflows,” *J. Mech. Des.*, vol. 136, no. August, Jun. 2014.
- [86] F. P. Brooks Jr., *The Mythical Man-month (Anniversary Ed.)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.
- [87] J. Mihm, C. H. Loch, D. Wilkinson, and B. a. Huberman, “Hierarchical Structure and Search in Complex Organizations,” *Manage. Sci.*, vol. 56, no. 5, pp. 831–848, May 2010.
- [88] J. Rajapaksha, K. Mirkovic, D. Robinson, and D. Wynn, “Modelling and simulating the effect of coordination on pd performance while handling change,” in *Proceedings of the International Conference on Engineering Design, ICED*, 2017, vol. 2, no. DS87-2.
- [89] V. Levardy and T. R. Browning, “An Adaptive Process Model to Support Product Development Project Management,” *Ieee Trans. Eng. Manag.*, vol. 56, no. 4, pp. 600–620, 2009.

This is a pre-print / author accepted version. Please cite article as:

Parraguez Ruiz, P., Piccolo, S., Perišić, M. M., Štorga, M., & Maier, A. (Accepted/In press). Process modularity over time: modelling process execution as an evolving activity network. *IEEE Transactions on Engineering Management*. <https://doi.org/10.1109/TEM.2019.2935932>.

- [90] A. MacCormack, J. Rusnak, and Baldwin, “Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code,” *Manage. Sci.*, vol. 52, no. 7, pp. 1015–1030, Jul. 2006.
- [91] M. Starnini, A. Machens, C. Cattuto, A. Barrat, and R. Pastor-Satorras, “Immunization strategies for epidemic processes in time-varying contact networks,” *J. Theor. Biol.*, vol. 337, pp. 89–100, 2013.