

# Performance Assessment in Self-organising Mechatronic Systems: A First Step towards Understanding the Topology Influence in Complex Behaviours

Pedro Neves<sup>1</sup>, Luis Ribeiro<sup>2</sup>, Mauro Onori<sup>1</sup>, and José Barata<sup>2</sup>

<sup>1</sup> KTH Royal Institute of Technology, Department of Production Engineering,  
Stockholm, Sweden

<sup>2</sup> CTS UNINOVA, Dep. de Eng. Electrotecnica, F.C.T. Universidade Nova de Lisboa,  
Monte da Caparica, Portugal  
{pmsn2,onori}@kth.se, {ldr,jab}@uninova.pt

**Abstract.** The research and development of self-organising mechatronic systems has been a hot topic in the past 10 years which conducted to very promising results in the close past. The proof of concept attained in IDEAS project [1] that plug&produce can be achieved in these systems opens up new research horizons on the topics of system design, configuration and performance evaluation. These topics need to consider that the systems are no longer static prototypes but instead several distributed components that can be added and removed in runtime. The distribution of modules in the system and their inherent connections will then potentially affect the system's global behaviour. Hence it is vital to understand the impact on performance as the system endures changes that affect its topology. This article presents an exploratory test case that shows that as a system evolves (and the nature of the network of its components changes) the performance of the system is necessarily affected in a specific direction. This performance landscape is necessarily complex and very likely nonlinear. Simulation plays therefore an important role in the study of these systems as a mean to generate data that can be later on used to generate macro level knowledge that may act as a guideline to improve both design and configuration.

**Keywords:** Multi-Agent Systems, Performance Assessment, Evolvable Production Systems, Self-Organizing Systems, Simulation.

## 1 Introduction

Assembly lines are flow oriented production systems which were originally built for cost-efficient mass production. Given the current turbulent markets they are now struggling to cope with constant re-design and re-configurations imposed by high customisation and low volumes [2]. As the business paradigm shifts towards an increased customization and personalization the requirements imposed on production change [3], [4]. Traditional systems present optimal solutions for particular forecasted products. However, they fail to present the desired agility to follow volatile market

needs. This augments the need of systems that are rapidly deployable, reconfigurable and autonomous and that can accommodate the required changes with minimal integration/programming effort and therefore maximum cost-effectiveness [5]. Modern production paradigms emerged in the last two decades aiming at offering responsive and cost effective solutions. Some examples are Bionic Manufacturing Systems [6], Reconfigurable Manufacturing Systems [7], Evolvable Production Systems (EPS) [8], Holonic Manufacturing Systems [9] and Changeable Manufacturing Systems [10]. They all share some core principles such as modularity, structure, heterogeneity, autonomy, interaction and dynamics [11]. All modern paradigms aim at the encapsulation of module functionalities as services to enable their seamless integration in production systems to tackle interoperability between systems and re-usability of legacy equipment. Modern paradigms exhibit a shift from centralized to distributed control architectures where each node of the system is capable of taking autonomous decisions and interacting with the other nodes on the network. This makes them suitable to handle aspects related to dynamic addition/removal of heterogeneous modules and change of products on the fly with little or none reprogramming efforts.

The Manufature roadmap [12] has pointed out the need of cross-sectorial research to develop adequate IT structures to support scalable and interoperable control systems, plug&play production modules and responsive factories through cooperative and self-organising control systems. Last decade's ICT research has been fruitful leading to architectures and IT middleware that support ready to use intelligence and autonomy at device-level. The most common approaches to implement distributed control automation are Multi-Agent Systems (MAS) and Service-Oriented Architectures (SOA). Given their loosely coupled nature they enable the dynamic composition of complex entities from simple services and support the dynamic plug and unplug of self-contained heterogeneous mechatronic modules [1], [11], [13], [14], [15].

These characteristics are desirable to increase system's responsiveness and tackle disturbances in the shop floor (e.g. machine failures, bottlenecks, volume and product change, etc.) [5], [16]. Nevertheless due to the early stage development of these systems, there is a lack of methods to support their design and configuration and to evaluate their behaviour although some analysis work has been carried out in [17], [18]. The purpose of this paper is therefore show a potential new system analysis path that may contribute to the design and configuration of complex manufacturing systems with a potential of generalization to collective systems. Particularly it provides a preliminary behavioural assessment of an EPS system to unveil the influence of system component's topology in the total make span. The subsequent details are organized as follows: section 2 frames this work under the scope of collective awareness systems; section 3 presents the state of the art of assembly line design, the research gap and the proposed analysis method; in section 4 it is detailed a preliminary experiment that motivates this research; and finally some concluding remarks are offered in section 5.

## 2 Relation to Collective Awareness Systems

The proliferation of embedded computing and web technology has opened a door for exploring distributed automation systems. In such a context, all entities in the shop floor are to some extent intelligent, able to reason and make decisions dynamically according to their context, forming dynamic loosely coupled networks of cooperative entities that exhibit collective behaviour. The main advantages of such approach are the increase of local autonomy to allow the addition/removal of equipment and improved robustness, fault tolerance and adaptability by empowering self-\* properties [1], [19].

The increase of IT infrastructures and IT middleware that support intelligence at device level also enables the collection of huge amounts of data concerning product and process design, logistics, assembly, quality control, scheduling, maintenance, fault detection, etc.; extending largely previous available data and adding more reliability to it [20]. Extracting knowledge from these operations is perceived as a tremendous opportunity to improve productivity and efficiency both at product and system levels which is important to reduce costs. This can contribute to the comprehension of the local-global relations in the system and the development of better system designs that can fully explore the self-organizing essence of modern paradigms. As a natural consequence, this would contribute to the development of collective awareness systems and allow enterprises to target volatile business opportunities and build the fundamental pillars for competitive sustainable innovation.

## 3 State of the Art and Problem Definition

Assembly line design has as main goal the maximization of the ratio between throughput time and required costs and incorporates all decisions concerning resource planning, sequence planning and system balancing [2]. These activities are often carried out by a human expert based on experience and know-how. Consequently many design alternatives are left unexplored and the quality of the line design will be highly dependent on the competence of designer [21]. Assembly Line Design (ALD) comprises two main sub-topics: Assembly Line Balancing (ALB) and Resource Planning (RP). ALB's more common problems consist in assigning tasks to workstations to satisfy a specific objective function while RP problems, on the other hand, consider the case when more than one type of equipment is available [2]. Considering the state of the art of ALD, one can find a vast number of contributions in literature regarding ALB. Especially regarding Simple Assembly Line Balancing (SALB) which introduces several simplifications and restrictions to the General Assembly Line Balancing (GALB) problem such as: homogeneous products, fixed cycle times, deterministic operation times of tasks and serial layout line. These simplified models often fail to reproduce the reality and therefore are rarely used in practical industrial applications [21], [22]. Given the early stage development of self-organising mechatronic systems little research has been conducted on assembly line design. Nevertheless, the different nature and goals of these systems in

comparison to traditional ones enables the identification of some aspects that might demand new methods: 1) modern paradigms do not target the design of prototypic tailored systems for a specific product but instead the design of responsive and cost-effective solution for evolving requirements; 2) the encapsulation of module functionalities as services and their seamless integration in production systems enables different equipment to be plugged and unplugged in runtime without reprogramming; 3) equipment functionality can be very specific or multi-purpose hence the layouts and network structure are fundamental; 4) modules are intelligent and the system behaves as a cooperative network of modules contrary to centralized automation approaches; 5) task assignment is performed dynamically following the current system status and there is no a-priori assignment of product tasks to specific equipment. The above mentioned points justify the need of a method that can present suitable real-life configurations to apply in Self-organizing mechatronic systems for scenarios of constant system evolution to better fit evolving production requirements. The overall behaviour of these systems results from the interactions and self-organization of its constituents which are prime characteristics of complex adaptive systems [23]. Furthermore as suggested in [24] the topology of the network can play major role in the self-organising phenomena present in random networks. The research question addressed by this work is:

- “How can the performance of an assembly system composed by a dynamic network of intelligent collaborative modules be evaluated?”
- “What methods can support the design and configuration of an assembly system composed by a dynamic network of intelligent collaborative modules?”

To address these questions a simulation tool was developed to generate data resulting from distinct system modifications and assess the impact of those changes in the system make span. Given the experimental nature of the work the classical research method was adopted.

## 4 Experiment Design

### 4.1 The Simulation Agents

In order to design and implement an experiment to study the influence of network topology on self-organising mechatronic systems the EPS paradigm was considered. A full description of Evolvable Production Systems principles and architectures can be found on [25], [26] and the references therein. The starting point of the current research is the generation of representative data. This data has to be necessarily generated by simulation, taking some basic EPS systems and allowing them to evolve. The test-case presented in this article has the goal of providing a first indication of whether network topological features can be used to characterize the assembly system composed by a network of modules or not. And more important if these features exhibit any relation with traditional performance metrics such as the make span. The preliminary experiment considered in this paper is based in the evolution of a U-shaped serial line. The original layout is transformed by adding more paths between

the stations in the system. The simulation environment was developed using the JADE framework [27] and all communications in the platform follow the FIPA-Request protocol [28]. The system has 5 basic agents that can be deployed:

- Station Agent – abstracts a module in the system where specific assembly processes can be performed (corresponds to a node in the network).
- Router Agent – abstracts a diverter and links conveyors together being responsible to route the carriers to different conveyors.
- Conveyor Agent – abstracts a conveyor with specific size and capacity controlling the flow of products.
- Carrier Agent – abstracts a carrier in the system responsible to carry the product and follow its production plan. The carriers queue in the Entry Point until they are allocated to a product, and then follow the requests of the product. When they receive requests from the product they are responsible to find the shortest path using Dijkstra Algorithm and inform the product when they reach a location where the process can be executed.
- Entry/Exit Point Agent – these agents are responsible for the entrance and exit of carriers in the system.
- Product Agent – abstracts a product and manages its production plan.

## 4.2 Experimental Setup

In the following experiment a setup with 6 stations with functionalities glue, stack, screw, pick&place, insert and weld is used. It is considered that all of them require the same time to execute their process (3 seconds). The positions of the stations in the system are fixed and only the paths connecting them change. All the conveyors have the same size (4m) and speed (1m/s). Consequently all the carriers travel at the same speed inside the conveyors and have the same size (0,2m). Conveyors are connected by routers and the transfer time is corresponds to a fixed time of one second. Upon request from the product to find a specific process in the system, the carrier will calculate the shortest path at each step of the move (i.e. at each new node) and send request messages to conveyors, stations and routers in order to go through them. Once it arrives to a station that can execute the requested process it sends an acknowledgment (inform message) to the product agent so it can request the skill execution to the station. One Entry Point agent and one Exit Point agent are also deployed where the carriers can enter and leave the system respectively. The considered base system for the tests is depicted in figure 1 (left). A total of 18 system evolutions (adding successive links) were considered and each one of these system was ran 4 times. Table 2 presents the network features extracted and the achieved make span results. Gephi [29] was used to extract the following system features:

- Average Degree Centrality – the centrality degree of a vertex is given by the total number of vertices adjacent to the vertex.
- Average Closeness Centrality – the closeness degree measures the degree to which a node is near all other nodes in a network. It is the inverse of the sum of the shortest distances between each node and every other node in the network.

- Average Betweenness Centrality – it measures to which extent a node is connected to other nodes that are not connected to each other.
- Average Cluster Coefficient – it corresponds to the measure of degree to which nodes in a graph tend to cluster together.

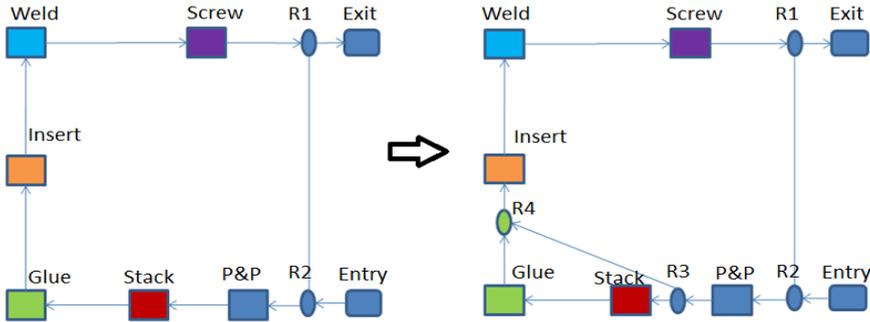


Fig. 1. Example of evolution from Network 1 to Network 2

A total of 12 carriers were deployed which are allocated to the products upon request, and then a total of 80 products from 8 different types (10 of each) were progressively deployed and associated with those carriers (Table 1).

Table 1. Products' workflows deployed

Product	Skill 1	Skill 2	Skill 3	Skill 4	Skill 5
PA1	Pick&Place	Glue	Insert	Weld	Screw
PA2	Stack	Glue	Insert	Screw	Weld
PA3	Insert	Weld	Screw	Glue	Screw
PA4	Stack	Screw	Pick&Place	Glue	Weld
PA5	Pick&Place	Insert	Screw	Weld	Stack
PA6	Glue	Insert	Screw	Weld	Stack
PA7	Insert	Glue	Screw	Weld	Pick&Place
PA8	Weld	Screw	Glue	Stack	Pick&Place

The products “compete” in the platform to be allocated and it is not guaranteed the order they enter the line and therefore results can vary substantially between executions as one can observe in the analysis of results in the next section.

### 4.3 Preliminary Result's Assessment

The obtained results are depicted in table 2 presenting the average system make span and the network features of each system evolution. It can be observed that as more paths are added in almost all cases the average system make span decreases. Nevertheless, the most significant leaps in performance gain are found in evolution 1→2, 2→3, and 6→7. In the other evolutions the performance is not significantly

improved which gives us an indication that exploring some paths can be more rewarding than others. The decrease of make span is somehow expected as the increase of paths results in shortest paths to different destinations and the influence of modules' positions is lowered and the travel time is minimized. Regarding the network features mentioned in section 4.2 one can observe they exhibit a consistent variability as more paths are added. The base system considered is a typical U-shaped serial line where each station has one inbound and one outbound and as more paths are added (more inbounds and outbounds to stations) the centrality degree of nodes increases. With the evolution of the system the average closeness decreases since this feature measures the proximity of a node to all nodes in the network.

**Table 2.** System features and achieved simulation results

<b>Sys.</b>	AVG degree centrality	AVG closeness centrality	AVG Between. centrality	AVG Cluster Coeff.	AVG Make Span	Standard deviation
<b>1</b>	1	2.81246	12	0	871,4335	30,5132
<b>2</b>	1.125	2.37798	9.25	0	802,8395	26,2787
<b>3</b>	1.25	2.21131	8.25	0	728,3610	4,4376
<b>4</b>	1.375	2.04464	7.25	0	712,9408	20,6743
<b>5</b>	1.625	1.8006	5.8112	0	693,1173	17,0078
<b>6</b>	1.75	1.71725	5.24999	0	696,9508	18,4620
<b>7</b>	1.875	1.675595	5	0	640,8648	16,1297
<b>8</b>	2	1.5774	4.375	0.108	625,1785	11,5933
<b>9</b>	2.125	1.5149	4	0.198	627,0783	11,2726
<b>10</b>	2.375	1.452	3.62501	0.333	618,2798	13,6728
<b>11</b>	2.5	1.411	3.375	0.367	625,6625	3,85056
<b>12</b>	2.625	1.36896	3.125	0.4	622,2423	9,8987
<b>13</b>	2.75	1.3481	3	0.458	620,0778	2,0295
<b>14</b>	2.875	1.3094	2.75	0.479	621,0133	20,7888
<b>15</b>	3.125	1.3304	2.5	0.485	611,1700	8,5325
<b>16</b>	3.25	1.3095	2.375	0.515	618,4785	5,4101
<b>17</b>	3.375	1.2262	2.2504	0.544	615,6785	14,6321
<b>18</b>	3.5	1.20525	2.125	0.56	605,8348	23,6286
<b>19</b>	3.625	1.1845	2	0.577	598,7913	11,4700

The average betweenness decreases with system iteration since it measures to which extent nodes are connected to other nodes that are not connected to each other. This makes sense since the connectivity is improving in the system and almost all nodes will be connected to each other. The average cluster coefficient also increases

as it measures how nodes cluster together (triplets of nodes). Regarding the existence of relations between these network features and the make span one can notice that they exist. As the average degree augments the make span is reduced which suggests that the connectivity of the nodes is an important feature to include in the system characterization. A similar conclusion can be deduced by analysing the average closeness of the nodes in the system. When it decreases the make span also decreases since it measures the degree to which a node is near all other nodes in a network. This means that the average closeness can also be potentially considered to characterize the system. The same conclusions can be taken from looking at both average betweenness and average clustering coefficient since they are varying consistently in one direction with the decrease of make span which makes them good candidates for system characterization as well.

This first test-case suggests that there are in fact relations between network features and the system make span that can be further studied. A limiting factor in this experiment was the use of only 12 carriers in all tests. The main goal of this experiment was to have a first indication if network metrics present a consistent variability that can be linked with the overall system performance that results from the interactions and self-organisation of modules. It was possible to conclude that this relation exists and can be further explored. If these relations are fully understood then we can realize the influence of size and topology in the self-organising process and potentially contribute to the generation of better system design and configuration. In highly pluggable systems most of the times the challenge is not to plug more devices but to understand which, where and how to plug these devices and the preliminary test case shows that some system evolutions are more significant than others. Hence the use of machine learning methods will be fundamental as it will enable us to relate all these variables and discover redundancies and dependencies between the features analysed and extract rules that map their real relations to the make span and other performance metrics. Since the relation between performance metrics and networks features is not necessarily linear, Rough Sets present a good candidate technique for this task since they are powerful in the discovery of redundancies and dependencies between the features of objects to be classified and therefore they represent a good approach to classification and rule extraction [30]. Rough set theory provides a mathematical approach using lower and upper approximations to deal with uncertain information and has proved to be useful in exploratory manufacturing applications for fault diagnosis [31], [32].

## 5 Concluding Remarks

To date most assembly design methods in use target optimized designs from scratch assuming constant the product mix and volumes and trying to match the needed processes in equipment in a repetitive way which is suitable mainly for large volumes and stable production. This research targets, on the other hand highly reconfigurable systems that can evolve autonomously such as the ones developed under the FP7 IDEAS project. One important point is therefore to guide the direction of this

evolution. On the other hand data collection from existing system will enable better initial designs. Therefore this research aims and understanding what are the relevant variables, from a system network perspective that can affect its overall performance. The preliminary test case suggests that there are correlations between the network characteristics and its performance. It also shows that certain evolutionary steps are much more relevant than others. The ability to evolve a system in the right direction using the correct "shortcuts" and avoiding mechanical or logical reconfigurations with a marginal gain is extremely important from a cost perspective. Simulation will play a decisive role in the generation of data that can be mined and reused for the assessment of what are the most important features and changes to consider in an EPS system. The present work is only the first step towards a far more ambitious goal. The next steps of this research include the investigation of the suitability of graphs with small-world properties to describe the systems' topology and perform tests by using the generation of random networks and evolutions in the simulation tool. Additionally the performance assessment context will be extended to include more indicators and there will be an incorporation of rule-extraction methods such as rough sets to help us dealing with massive amounts of data and bridge these two dimensions.

## References

1. Onori, M., et al.: The IDEAS Project: Plug & Produce at Shop-Floor Level. *Assembly Automation* 32(2), 4–4 (2012)
2. Boysen, N., Fließner, M., Scholl, A.: A classification of assembly line balancing problems. *European Journal of Operational Research* 183(2), 674–693 (2007)
3. Maffei, A.: Characterisation of the Business Models for Innovative, Non-Mature Production Automation Technology, KTH (2012)
4. Hu, S.J., et al.: Assembly system design and operations for product variety. *CIRP Annals-Manufacturing Technology* 60(2), 715–733 (2011)
5. Onori, M., Barata, J.: Outlook report on the future of European assembly automation. *Assembly Automation* 30(1), 7–31 (2009)
6. Ueda, K.: A concept for bionic manufacturing systems based on DNA-type information. In: 8th International PROLAMAT Conference on Human Aspects in Computer Integrated Manufacturing, North-Holland Publishing Co. (1992)
7. Koren, Y., et al.: Reconfigurable manufacturing systems. *CIRP Annals-Manufacturing Technology* 48(2), 527–540 (1999)
8. Onori, M., Barata, J.: Evolvable Production Systems: New domains within mechatronic production equipment. In: IEEE International Symposium on Industrial Electronics (ISIE). IEEE, Bari (2010)
9. Valckenaers, P., et al.: Holonic manufacturing systems. *Integrated Computer-Aided Engineering* 4(3), 191–201 (1997)
10. Wiendahl, H.P., et al.: Changeable manufacturing-classification, design and operation. *CIRP Annals-Manufacturing Technology* 56(2), 783–809 (2007)
11. Ribeiro, L., et al.: MAS and SOA: A Case Study Exploring Principles and Technologies to Support Self-Properties in Assembly Systems. In: SARC 2008: Self-Adaptation for Robustness and Cooperation in Holonic Multi-Agent Systems, p. 43 (2008)
12. Jovane, F., Westkamper, E., Williams, D.: *The ManuFuture Road: Towards Competitive and Sustainable High-Adding-Value Manufacturing*. Springer (2008)

13. Jammes, F., Smit, H.: Service-oriented paradigms in industrial automation. *IEEE Transactions on Industrial Informatics* 1(1), 62–70 (2005)
14. Shen, W., et al.: Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced Engineering Informatics* 20(4), 415–431 (2006)
15. Marik, V., Lazansk, J.: Industrial applications of agent technologies. *Control Engineering Practice* 15(11), 1364–1380 (2007)
16. Ribeiro, L., Barata, J.: Re-thinking diagnosis for future automation systems: An analysis of current diagnostic practices and their applicability in emerging IT based production paradigms. *Computers in Industry* 62(7), 639–659 (2011)
17. Ribeiro, L., Rosa, R., Barata, J.: A structural analysis of emerging production systems. In: 10th IEEE International Conference on Industrial Informatics (INDIN). IEEE (2012)
18. Farid, A.: An Axiomatic Design Approach to Non-Assembled Production Path Enumeration in Reconfigurable Manufacturing Systems. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC 2013), Manchester, UK (2013)
19. Ribeiro, L., et al.: Self-organization in automation-the IDEAS pre-demonstrator. In: 37th Annual Conference of the IEEE Industrial Electronics Society (IECON 2011). IEEE, Melbourne (2011)
20. Harding, J.A., Shahbaz, M., Kusiak, A.: Data mining in manufacturing: a review. *Journal of Manufacturing Science and Engineering* 128, 969 (2006)
21. Michalos, G., Makris, S., Mourtzis, D.: An intelligent search algorithm-based method to derive assembly line design alternatives. *International Journal of Computer Integrated Manufacturing* 25(3), 211–229 (2012)
22. Boysen, N., Fliedner, M., Scholl, A.: Assembly line balancing: which model to use when? *International Journal of Production Economics* 111(2), 509–528 (2008)
23. McCarthy, I.P., Rakotobe-Joel, T., Frizelle, G.: Complex systems theory: implications and promises for manufacturing organisations. *International Journal of Manufacturing Technology and Management* 2(1), 559–579 (2000)
24. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. *Science* 286(5439), 509–512 (1999)
25. Ribeiro, L., et al.: Evolvable Production Systems: An Integrated View on Recent Developments. In: 6th International Conference on Digital Enterprise Technology. IEEE, Hong Kong (2009)
26. Onori, M.: Evolvable Assembly Systems - A New Paradigm? In: 33rd International Symposium on Robotics Stockholm (2002)
27. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing multi-agent systems with JADE. Wiley (2007)
28. FIPA, The foundation for intelligent physical agents (2008), <http://fipa.org>
29. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks. In: ICWSM (2009)
30. Pawlak, Z.: Rough set theory and its applications to data analysis. *Cybernetics & Systems* 29(7), 661–688 (1998)
31. Kusiak, A.: Rough set theory: a data mining tool for semiconductor manufacturing. *IEEE Transactions on Electronics Packaging Manufacturing* 24(1), 44–50 (2001)
32. Peng, J.-T., Chien, C.-F., Tseng, T.: Rough set theory for data mining for fault diagnosis on distribution feeder. In: IEE Proceedings- Generation, Transmission and Distribution, IET (2004)