Adaptive Scheduling based on Self-organized Holonic Swarm of Schedulers

Paulo Leitão[1,2], José Barbosa[1]

[1] Polytechnic Institute of Bragança, Campus Sta Apolónia, Apartado 1134, 5301-857 Bragança, Portugal
{pleitao, jbarbosa}@ipb.pt
[2] Artificial Intelligence and Computer Science Laboratory, Rua Campo Alegre 102, 4169-007 Porto, Portugal

Abstract- **Scheduling plays an important role in the companies' competiveness, dealing with complex combinatorial problems subject to uncertainty and emergence. In particular, in the ramp-up phase of small lot-sizes of complex products, scheduling is more demanding, e.g. due to late requests and immature technology products and processes. This paper presents the principles of a distributed scheduling architecture based on holonic and swarm principles and implemented using multi-agent system technology. In particular, it is described the coordination among the network of the swarm of schedulers and analysed the impact of embedded self-organization mechanisms.**

I. INTRODUCTION

In spite of fluctuations and uncertainties provoked by the effects of the 2009's crisis and the reallocation of production sites to emergent countries with cheaper labour force, the control and monitoring domain worldwide has a market value of 190 billion Euros [1], and within this, the sum of automotive, manufacturing and process industries represents 60% of the total market [1]. Additionally, according to [2], "*Manufacturing is the largest industry market segment for enterprise IT and offers the largest potential market for cloud computing and software as a service*". In this context, planning and scheduling systems assume a critical importance to reach the competitiveness levels of an enterprise placed in the current worldwide market, providing decision support concerning tactical and operational planning, scheduling and real-time optimization methods. The ability to create and adjust long-term plans and short-term schedules according to the production changes, availability of resources and requests from the customers is a key factor for success.

Scheduling refines general plans coming from the strategic planning level and elaborate detailed workshop schedules. The scheduling is a complex problem mainly due to its highly combinatorial aspect and its dynamic nature [3]. Traditionally, optimization algorithms are used to solve these problems considering them as static and deterministic. However, the dynamic and stochastic nature of industrial environments increase the problem complexity, demanding more adaptive and efficient handling of either planned or unplanned events in real time, such as changing volume of orders, high priority orders, resources breakdown, worker's illnesses and large repairs. Also important to be considered is the complexity associated to the ramp-up phase of complex and highly customized products, which are exceptionally challenging for planning, scheduling and control, especially in small lot sizes. In fact, daily challenges, like immature high technology products and processes, and late requests for change, create significant constraints. Having this in mind, new methods and Information and Communication Technologies (ICT) are required to develop mitigation strategies to respond faster to unexpected events, creating new, emergent approaches for the scheduling problem.

The objective of this work is to design these adaptive scheduling systems using multi-agent systems as the generic solid basis for developing complex Systems of Systems (SoS) as swarms of schedulers using the inspiration from biology and theory of complexity. In fact, the use of swarm principles, inherited from biology, simplifies the system design by organizing in a holonic perspective multiple workshop/factory scheduling systems to achieve easier the coordination of interdependent schedules over the enterprise. The proposed approach goes beyond the state of the art with the idea of holonic networks of multi-agent swarm of real time schedulers, showing self-organized systems that replace the traditional waterfall scheduling schemes based on "master-slave" relations.

The rest of the paper is organized as follows. Section II states the problem addressed in this work and Section III introduces the architectural approach to dynamic, adaptive scheduling based on swarm principles applied to multi-agent systems. Section IV describes the application of the holonic perspective to the swarm of schedulers architecture aiming an easier coordination among interdependent schedules and Section V introduces the self-organization mechanisms to regulate the dynamics in such network of schedulers. At last, Section VI rounds up the paper with the conclusions.

II. PROBLEM STATEMENT

The scheduling problem has been widely studied and referred in literature, being defined as the optimal allocation of resources to jobs over the time, where these assignments must obey to a set of constraints that reflect the temporal relationships between jobs and the capacity limitations of the resources.

Small scheduling problems can be solved using mathematical programming, such as linear programming, to obtain optimal solutions. However, the manufacturing scheduling becomes a complex combinatorial problem, more specifically Non-deterministic Polynomial-time hard (NP-hard) problems, even for relatively small scheduling problems. In this case, these algorithms are not adequate since they may require a huge amount of time to compute the optimal solution. Since the experience shows that usually it is not important to have the best solution but a satisfactory fast solution, the alternative is to use other methods like neighbourhood search techniques (e.g., simulation annealing or taboo search) or population optimization strategies (e.g. Particle Swarm Optimization or Genetic Algorithms).

The previous scheduling strategies consider the manufacturing scheduling as static and deterministic. However, the industrial manufacturing scheduling is subject to a dynamic environment, with new jobs arriving continuously to the system, certain resources becoming unavailable and additional resources introduced. The reactive distributed scheduling deals with the dynamic and stochastic nature of the problem. The distributed scheduling means that the scheduling algorithm is distributed over a number of distributed entities that combine their calculation power and their local knowledge to optimise the global performance [4]. The major advantages of the distributed scheduling are the improvement of reaction to disturbances and the parallel computation. Agent technology is being used to solve the manufacturing scheduling problem, using these distributed principles, and mainly using algorithms based on centralised techniques, or other based on emergent behaviour, such as market-based or contract net protocol. Examples of these types of manufacturing scheduling are [3], [5] and [6].

The major disadvantages of the available state-of-the-art systems for the manufacturing scheduling problem are their limitations to re-schedule the whole schedule, which becomes very time consuming and impossible to be achieved in real-time, and the missing scalability to handle the simultaneous deployment at multiple sites. Even the use of multi-agent systems is not always enough to face the growth of complexity and scale of production.

III. Swarm of Schedulers

The described challenge can be faced by getting inspiration from biology, and particularly swarm and self-organization concepts, translating the principles found in biology to design planning and scheduling solutions. Note that the idea is to translate these concepts and not simple copy them.

A. Foundation principles

Nature and biology offer a plenty of powerful mechanisms to handle emergent and evolvable environments, where complex systems are built upon entities that exhibit simple behaviours and have reduced cognitive abilities. An amazing example is the concept of swarm, found in colonies of insects, which is based on the following basic principles:

- Community of entities regulated by very simple rules.
- Mainly reactive behaviour, i.e. limited learning capabilities.
- Entities interact with each another and with their environment.
- No central authority.

Swarm intelligence is a derived concept that can be defined as "*the emergent collective intelligence of groups of simple and single entities*" [7], reflecting the emergent phenomenon, which occurs without a pre-defined plan, not driven by a central entity, and only when the resulted behaviour of the whole is greater and much more complex than the sum of the behaviours of its parts [8]. Some illustrative examples of this kind of behaviour can be found on the societies of ants and bees. In fact, everybody knows that "*a single ant or bee isn´t smart but their colonies are*" [9], and also that they are capable of exhibiting very surprising complex behaviours. In such environments, the coordination of activities uses simple feedback coordination mechanisms in opposite to the traditional rigid and monolithic centralized control.

B. Architectural concept

The proposed architectural schema, for more adaptive and efficient scheduling solutions, considers the swarm principles in the design of these tools in the manufacturing domain. In the proposed approach, each scheduling system is a network of schedulers, organized as a swarm of schedulers, as illustrated in Fig. 1, each one being responsible for the schedule of a production organizational unit of an enterprise, namely a factory, workshop or station.

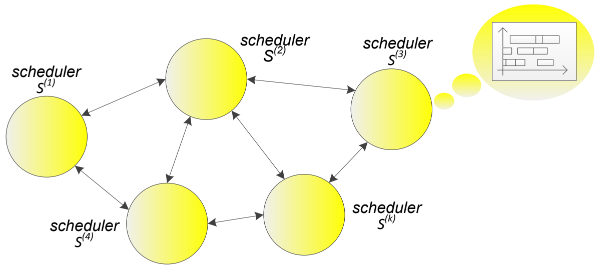

Fig. 1. Scheduling system as a swarm of schedulers

Definition 1. A swarm of schedulers is a network of individual schedulers, $S = \{S^{(1)}, S^{(2)}, \dots, S^{(k)}\}$, organized as a swarm, where the global schedule E emerges from all the non-linear interactions, R= $\{S^{(m,n)}, m, n \in \{1,..,k\}\}$, among these schedulers, $E = (S, R)$.

The individual schedulers are constantly interacting with each other to achieve the global (emergent) scheduling, exchanging information on the events that occur during the production process and could affect the order execution of the others. The swarm of schedulers is connected to the legacy systems, namely database and ontology services, enabling the schedulers to retrieve required data from these sources.

When considering the design of adaptive scheduling systems, as systems of systems based on a swarm of schedulers, a pertinent question is how to define swarms to balance interests of selfish individuals and groups of individuals. Two distinct strategies can be implemented:

- The scheduling problem may be divided into several smaller scheduling problems according to a proper logical dependency, e.g. a workshop scheduler can comprise several station schedulers.
- The scheduling problem may be divided in a way that different schedulers can be used to search faster alternative solutions in parallel, e.g. using different scheduling algorithms or exploratory searches.

Both situations reinforce the importance of using the swarm principles in the design of these scheduling systems.

The specification of individual schedulers is independent of the swarm principles and can be implemented using different algorithms. Indeed, a scheduler can be implemented through commercial software that runs an optimization algorithm, such as ILOG™ running CPLEX/MILP, or a piece of software implementing an optimization technique, such as simulating annealing or genetic algorithms. Also, individual schedulers can be implemented using multi-agent technology, composed by a set of software agents that represent interests of orders, resources, workers, operations and materials. In the last approach, the agents interacts each other to achieve the overall schedule through proper negotiation mechanisms.

Definition 2. An individual scheduler, $S^{(i)}$, may be composed by a network of entities, $S^{(i)} = \{S_1^{(i)}, S_2^{(i)}, \dots, S_{ik}^{(i)}\}$, which are able to produce the desired scheduling and where entities are related by a set of relations $R^{(i)} = \{S_{jl}^{(i)}, j, l \in \{1, .., i_k\}\}$. In case of a centralized technique, the network has a singular dimension, being $S^{(i)} = \{S_1^{(i)}\}$ and $R^{(i)} = \{\ \ \}$.

Depending of homogeneity of individual schedulers in terms of similarity, two different types of swarm can appear, as illustrated in Fig. 2:

a) *Swarm of homogeneous schedulers*, i.e. all individual schedulers have the same internal behaviour.

b) *Swarm of heterogeneous schedulers*, i.e. individual schedulers may have different internal behaviour. Note that in this case, the entities may be similar in terms of objectives but using different architectural principles.

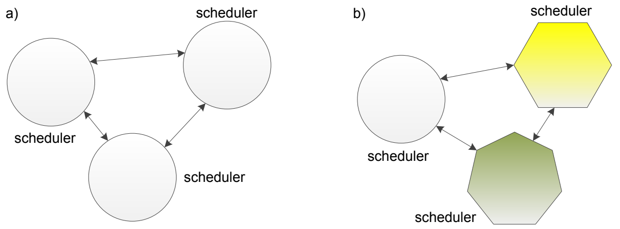

Fig. 2. Swarm of: a) homogeneous schedulers, b) heterogeneous schedulers

The use of homogeneous or heterogeneous schedulers is strongly dependent of the application domain and the design strategy, but in the proposed approach, the swarms of heterogeneous schedulers are supported in a transparent manner, contributing for a smooth migration from old solutions. For this purpose, the proper definition of communication interfaces is required, as the mean for data exchange between individual schedulers. Naturally, and additionally for more complex scenarios, it is also necessary the development of customized behaviours within the schedulers. An example can be seen when an individual scheduler receives a request, from another one, to verify the possibility to accommodate some changes in the previous one. Note that this is not a disturbance but a what-if situation that is not normally treated in scheduling.

C. Coordination among individual schedulers

An important question associated to the design of swarm of schedulers is the coordination between the independent schedulers inside the swarm to achieve the global scheduling.

Being organized in a network of swarm of schedulers, these entities coordinate their activities to produce a feasible consistent schedule. The events are first processed locally by individual schedulers and propagated further into the network, only if necessary. The system should produce the schedule with a reasonable consistency, i.e. ensuring the coherence among the several individual schedulers.

This coordination is based on feed forward and feed backward propagation of events among individual schedulers, as illustrated in Fig. 3.

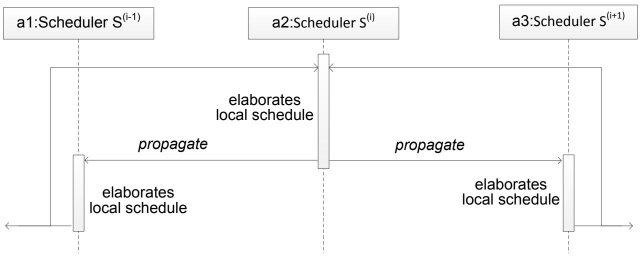
a1:Scheduler S(i-1)
a2:Scheduler S(i)
a3:Scheduler S(i+1)
elaborates local schedule
propagate
propagate
elaborates local schedule
elaborates local schedule

Fig. 3. Interaction pattern among schedulers inside a swarm of schedulers

In the figure, the scheduler $S^{(i+1)}$ represents all individual schedulers that have posterior dependencies from scheduler $S^{(i)}$ and the scheduler $S^{(i-1)}$ represents all the schedulers that may have influence on scheduler $S^{(i)}$. The propagate message refers to deviations occurred during the (re-)scheduling process in the scheduler $S^{(i)}$, e.g., delays or anticipations, which will affect/impact the other schedulers.

Definition 3. In a swarm of schedulers, each individual scheduler is connected to some precedent schedulers and posterior schedulers, according to the scheduling logic, $P = \{P^{(1)}, \dots, P^{(k)}\}$, propagating feed forward and feed backward the scheduling events accordingly with this set of precedencies. Note that this precedence issue also appears inside each individual scheduler,$S^{(i)}$, with the same logic, $P^{(i)} = \{P_1^{(i)}, \dots, P_{ik}^{(i)}\}$.

A crucial question is related to how individual schedules emerge into a global schedule. Particularly, it is important to guarantee that the emergent global schedule is desirable (convergent), i.e. not follow a black swan (note that the local optimal in each individual scheduler doesn't mean the optimal in the global scheduler). This can be ensured by using evolutionary mechanisms based on adaptation, evolution and selection that happen in a natural context, e.g. relaxing some critical processes or by selecting best scheduling candidates as it happens in Genetic Algorithms. Additionally, "small" events or deviations can trigger unexpectedly big changes in other schedulers, recalling the butterfly effect, which are considered as "unstable equilibriums" propagating through the whole network of schedulers. This "wave of changes" may be limited by applying market-based mechanisms, e.g. using bonuses and penalties as reward incentives.

In another level of concern, each individual scheduler can, in its idle time, schedule, in a shadow way, i.e. without practical effects, using different scheduling algorithms. The output of this simulation game would be used as a way to determine if the scheduler, using a different algorithm, would achieve other, better or worse, results. Based on these results, internal rewards mechanisms are used to assess the scheduler about the most effective optimization technique. This creates a ranking based table for the selection of the optimization algorithm accordingly.

The previous, micro-level behaviour, will enable that most critical schedulers, e.g. those where a higher number of disturbances appear, can dynamically select the most appropriate optimization algorithm and the adjacent schedulers to those, to use algorithms that provide more backslash to accommodate possible delays.

IV. Holonic Swarm of Schedulers

In a horizontal dimension, the scheduling systems follow the swarm principles. However, a pertinent question appears when it is necessary to handle different hierarchical levels of schedulers (or in other words, how the scheduling problem is divided into smaller scheduling problems). In fact, scheduling systems can be seen under a hierarchical scope having different levels of granularity, namely:

- *Factory level*, e.g. the Airbus plants of Hamburg and Toulouse.
- *Workshop level*, e.g. the assembly line in the Hamburg plant to assembly the fuselage.
- *Station level*, e.g. the stations 88 and 85 in the assembly line.
- *Team level*, e.g. the different teams in the station 88.
- *Worker level*, e.g. the different workers of one team of the station 88.

In a same manner, the network of schedulers can include schedulers on the same level of process details (with horizontal negotiations among them) and schedulers responsible for different time horizons and levels of details (e.g. operational, tactical and strategic level, requiring a vertical negotiation among them).

The different swarm of schedulers developed for the several granularity levels can be simply modelled by using the holonics principles.

A. *Basics on holonics*

Holonics is a paradigm introduced by the philosopher Arthur Koestler to living organisms and social organisations, mainly those that are complex hierarchical systems formed by intermediate stable forms, based on Herbert Simon theories and on his observations. Simon observed that complex systems are hierarchical systems formed by intermediate stable forms that do not exist as auto-sufficient and non-interactive elements but, on the contrary, are simultaneously a part and a whole. Koestler concluded that parts and wholes do not exist in domain of life, and proposed the word holon to represent this hybrid nature [10].

Holons combine the whole and the part, being simultaneously self-contained wholes to their subordinated parts, and dependent parts when seen from higher levels (Janus effect), as illustrated in Fig. 4. This property permits to break a holon into several others holons, which in turn can be broken into further holons, allowing the reduction of the problem complexity. A holon possesses two important characteristics: autonomy, where the stability of the holons result from their ability to act autonomously in case of unpredictable situations, and cooperation, which is the ability to have holons cooperating, transforming these holons into effective components of bigger wholes.

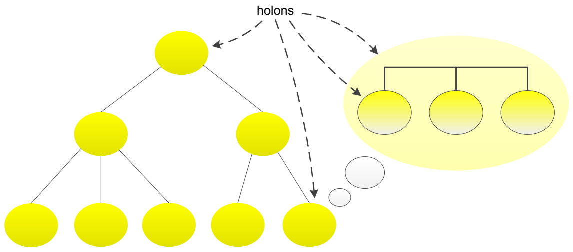

Fig. 4. Concept of holon, as a part and the whole

A holarchy is then defined as a system of holons, organised in a hierarchical structure, cooperating to achieve the system goals, by combining their individual skills and knowledge. The holons can integrate themselves into a holarchy and, at the same time, to preserve their autonomy and individuality, as well they can dynamically belong to multiple holarchies at the same time, which is an important difference to the traditional concept of hierarchies.

Several holonic approaches were proposed for manufacturing control, namely PROSA that combines the predictability and the robustness of the hierarchical control with the high reaction to disturbances of heterarchical control [11]. Another example is ADACOR [12] that addresses the agile reaction to condition changes by introducing an adaptive control approach that dynamically balances between a more centralised structure and a more decentralised one, allowing combining the global production optimisation with agile reaction to unexpected disturbances.

B. *Holonic perspective on swarm of schedulers*

The holonic principles were already applied in production planning (e.g. see [13] and [14]), and in manufacturing scheduling (e.g. [5] and [15]). In this work, the holonics principles allow the design of planning and scheduling systems as dynamic organizations of swarm of schedulers. These holonic structures appear when:

- Different temporal scopes are considered, i.e., one holon for each time horizon level, namely strategic, tactical and operational scheduling; each holon may comprise a swarm of schedulers.
- Different levels of granularity are considered, i.e., one holon for each hierarchical level of the scheduling problem, e.g. workshop holon, that may comprise several other holons for each one of the entities of the lower hierarchical level, e.g. several station holons; note that the emergent swarm of schedulers is also an holon.

In such approach, each scheduler is composed by a swarm of schedulers (i.e. homogeneous or heterogeneous entities, which may implement different methods or algorithms) and may be simultaneously part of a swarm of schedulers (i.e. recalling the Janus effect of the whole and the part inherent to the holonics principles), as illustrated in Fig. 5. Additionally, higher level holons (e.g. swarm of schedulers) can have a supervisory participation in lower level holons (e.g. individual schedulers) and a cooperation participation in same level within the holarchy.

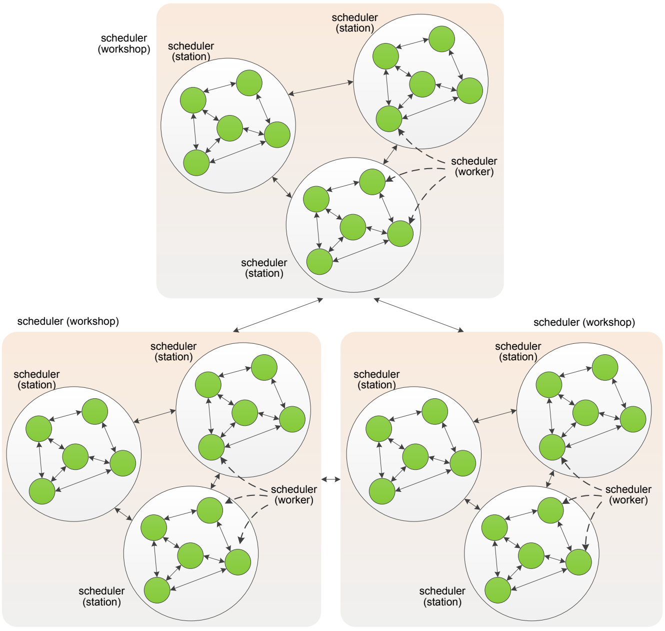

Fig. 5. Swarm of schedulers in a holonic perspective

In the proposed architecture, the schedulers are organized as swarm of schedulers forming holons organized as holarchies (i.e. a system comprises several hierarchical layers of schedulers). The coordination patterns among the swarm of schedulers should follow the principle "*as local as possible and as global as needed*", minimizing the number of interactions outside holarchies. This organization simplifies the cooperation and coordination among the schedulers, reducing the overall communication over the network.

The holonic swarm of schedulers apply proper mechanisms for finding optimal matching of demands and supplies (e.g. using market-based negotiation mechanisms or back-forward propagation). Since the swarm of schedulers are organized in a holarchy (following a hierarchy of schedulers), multiple scheduling systems (located at shop floor, assembly line, workshop or station levels) can cooperate in a network to achieve coordination of interdependent schedules over the factory. The cooperation mechanisms are performed by the interaction among the schedulers, which depends of granularity level, namely inside the individual schedulers or among swarm of schedulers. In case of interaction among different hierarchical levels of schedulers and with other systems, the interaction can use service-orientation platforms, namely an enterprise service bus (ESB), to overcome the interoperability problems.

V. Self-Organization

The design of such adaptive and self-reconfigurable scheduling systems also considers self-organization concepts to support the regulation of the dynamics of such complex network of swarm of schedulers.

As defined in the ADACOR^2 architecture [16], self-organization can be seen in two interconnected perspectives (see Fig. 6): structural (changing the relationships among the individual entities) and behavioural (changing the internal behaviour of individual entities).

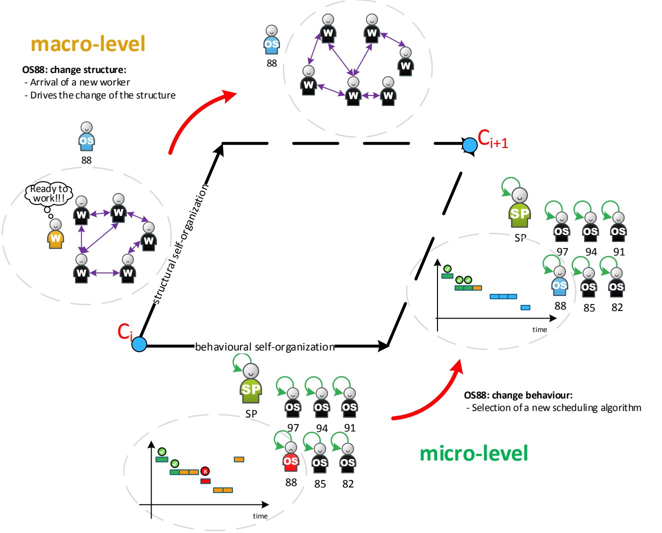
macro-level
OS88: change structure:
- Arrival of a new worker
- Drives the change of the structure
OS
88
W
Ready to work!!!
structural self-organization
C_i
C_{i+1}
behavioural self-organization
SP
97
94
91
88
85
82
time
OS88: change behaviour:
- Selection of a new scheduling algorithm
micro-level

Fig. 6. Self-organization in the ADACOR^2 architecture.

Definition 4. Following the ADACOR^2 principles, each holon has a set of behaviours $B = \{B_1, B_2, \dots, B_r\}$ and a set of structural relationships $St = \{St_1, St_2, \dots, St_a\}$ that guides the holon selection of appropriate scheduling algorithms and swarm associations.

In the context of the proposed architecture, behavioural self-organization can be seen in situations where individual schedulers change their algorithms or parameters to achieve better solutions (e.g., a long-term planning system that is changing or relaxing the problem constraints to achieve the solution faster). In other perspective, structural self-organization appears when:

- A dynamic reconfiguration of the scheduling problem occurs, e.g. by adding/removing workshops, stations or workers, or by changing the dependencies between the individual schedulers inside the swarm scope.
- A dynamic reorganization of individual entities, involved in exploratory searches for planning or scheduling solutions, occurs, aiming to achieve better emergent solutions.

The application of these self-organization principles leads to some important issues. Namely, these holarchies of swarm of schedulers can evolve dynamically, accommodating changes in the structures, dependencies and also in the behaviour of individual entities. This dynamic evolution based on the self-organization principles can cause some instability, being necessary to implement nervousness control mechanisms that allow pushing the system into its limits but maintaining in a stable state. A suitable approach, proposed by ADACOR^2, is a nervousness mechanism [16] based on the theory of control, and particularly in the PID (Proportional, Integrative and Derivative) control. This mechanism, applied into the scheduling context, allows the selection of proper scheduling algorithms and the discard of solutions that don't bring real benefit into the overall scheduling, maintaining the system stability. For this purpose, the nervousness mechanism is composed by three parameters: the component *P* is related to how fast the scheduler must react after the appearance of the perturbation, the parameter *I* is the improvement of the actual scheduling that can be accepted as a new solution, and the parameter *D* is the time to obtain the final scheduling, after which the current scheduling is accepted.

VI. Conclusions

The current demand of innovative scheduling systems, facing the frequent occurrence of unpredictable events (e.g., demand and resource fluctuations, new orders and resource unavailable), on-going innovations and fast changes of products, technologies and equipment, and the ramp-up processes, is critical for manufacturing enterprises placed in a worldwide market.

The proposed vision for the scheduling of large-scale manufacturing systems is the organization of such systems as a holonic multi-agent system network of swarm of schedulers, each one possessing an autonomous behaviour. In this approach, e.g., a workshop scheduler could be decomposed in an adaptive network of schedulers of stations, and successively, each station scheduler can be decomposed in a network of team schedulers. Such really holonic ("Russian matryoshka doll style") architecture contributes to provide very high openness, flexibility, scalability and reliability of fully distributed, intelligent large-size systems. Additionally, this approach allows the replacement of the traditional enterprise resource planning operating mainly in batch mode with daily-weekly-monthly cycles of scheduling, by a more efficient and adaptive real-time event-driven system.

Future work will be devoted to the application of the proposed holonic swarm of schedulers approach to develop planning and scheduling solutions for Airbus and Iacobucci use cases facing the ramp-up production of small lot sizes.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7 ARUM project, under grant agreement n° 314056.

References

[1] Eurostat, "Euro Indicators - Industrial Production", News Release 24/2012, Eurostat Press Office, December 2011.

[2] K.F. Brant, "Report Highlight for Market Trends: Cloud Computing and SaaS Adoption in Manufacturing and Natural Resources, Worldwide, 2012", Gartner, 2012.

[3] W. Shen, and D. Norrie, "An Agent-based Approach Dynamic Manufacturing Scheduling", Proceedings of the Workshop in Agent-based Manufacturing Workshop at Autonomous Agents, 1998.

[4] L. Bongaerts, "Integration of Scheduling and Control in Holonic Manufacturing Systems", PhD thesis, Katholieke Universiteit Leuven, Belgium, 1998.

[5] L. Gou, P. Luh, Y. Kyoya, "Holonic Manufacturing Scheduling: Architecture, Cooperation Mechanism, and Implementation", *Computers in Industry*, 37(3), pp. 213–231, 1998.

[6] P. Leitão and F. Restivo, "A Holonic Approach to Dynamic Manufacturing Scheduling", Robotics and Computer-Integrated Manufacturing, vol. 24, nº 5, pp. 625-634, 2008.

[7] E. Bonabeau, M. Dorigo and G. Theraulaz, "*Swarm Intelligence: from Natural to Artificial Systems*",Oxford University Press, 1999.

[8] J. Holland, "*Emergence: from Chaos to Order*", Oxford University Press, 1998.

[9] P. Miller, "The Genius of Swarms", *National Geographic*, July, 2007.

[10] A. Koestler, "*The Ghost in the Machine*", Arkana Books, London,1969.

[11] H. V. Brussel, J. Wyns, P. Valckenaers L. Bongaerts, "Reference Architecture for Holonic Manufacturing Systems: PROSA", *Computers In Industry*, 37(3), pp. 255-274, 1998.

[12] P. Leitão, F. Restivo, "ADACOR: A Holonic Architecture for Agile and Adaptive Manufacturing Control", *Computers in Industry*, 57 (2), pp. 121-130, 2006.

[13] T.Hasegawa, L. Gou, S. Tamura, P. B.Luh and J.M. Oblak, "Holonic Planning and Scheduling Architecture for Manufacturing", International Working Conference on Cooperating Knowledge Based Systems, Keele, U.K., pp. 125-139, 1994.

[14] B. Denkena, H. K. Tonshoff, M. Zwick and P.Woelk, "Process Planning and Scheduling with Multi-agent Systems", Knowledge and Technology Integration in Production and Services: Balancing Knowledge and Technology in Product and Service Life Cycle, pp. 339-348, 2002.

[15] P. Sousa and C. Ramos, "A Distributed Architecture and Negotiation Protocol for Scheduling in Manufacturing Systems", *Computers in Industry*, 38(2), pp. 103-113, 1999.

[16] J. Barbosa, P. Leitão, E. Adam and D. Trentesaux, "Structural Self-organized Holonic Multi-Agent Manufacturing Systems", Proc. of the 6th Int. Conf. on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS'13), LNCS, vol. 8062, Springer, pp. 59-70, 2013.