# Reactive Business Processes for Factory Automation

Domnic Savio, Stamatis Karnouskos, Luciana Moreira Sá de Souza, Vlad Trifa, Dominique Guinard, Patrik Spiess
SAP Research (www.sap.com)
Contact: domnic.savio@sap.com

***Abstract*—Modern enterprises operate on a global scale and depend on complex business processes. Business continuity needs to be guaranteed, while changes at the shop floor should happen on-the-fly without stopping the production process. Unfortunately, the existing business processes found in most enterprises are not modular enough, nor they have dynamic support from the device level. However, as the number of sophisticated networked embedded devices in the shop-floor increases, SOA concepts can now be pushed down and provide a better collaboration between the business systems and the production line. This leads to highly dynamic systems that can adapt and optimize their behavior to achieve their goals. The work presented here shows directions to achieve this dynamism by means of simulation, state identification and close coupling of real world and business systems.**

## I. Introduction

Business processes in a company are defined by the best practices of the respective industry and its goals. However, in reality production processes are mostly monolithic and expect results to be ideal. A production process usually has a series of vertical integrations towards the shop floor until end of the process lifetime. As a consequence, **challenges** arise when trying to make the processes adaptive or trying to extend it. Introducing variables to adapt to the dynamic nature of the shop floor is very expensive for companies that span multiple production locations and several heterogeneous IT systems.

The device world is changing drastically, as technology rapidly advances, the shop-floor becomes populated with highly sophisticated networked embedded devices that have faster CPU's, are more economical, more compact and go beyond being task specific. Such devices can provide their functionality as a service and play an active role. They go beyond controlling local loops and can provide tools that offer real-time analysis, as well as adaptive Graphical User Interfaces (GUIs) for operators. Such devices are already emerging in the market e.g. the CX1020 series Programmable Logic Controller (PLC) of Beckhoff [1]. Other companies like Schneider Electric, Siemens and ABB are experimenting in R&D projects (www.socrades.eu) and expected to create products in the short term.

Business process can take full advantage of such sophisticated devices by easily integrating them via Service Oriented Architecture approaches. As web services are suitable and capable of running natively on embedded devices, they provide an interoperability layer that leads to easier coupling with other components despite of their high heterogeneity. Device Profile for Web Services (DPWS [2]) and OPC-UA [3] are two of the emerging technologies for realizing web service enabled devices. Eventually, we are heading towards a fully service enabled and highly dynamic infrastructure. Due to the loose coupling nature of web services, compositions of services can be easily created to match the desired scenario, almost like one would create web mashups. Such compositions contain services running on the device, on the network and on the enterprise system layers as depicted in Figure 1.

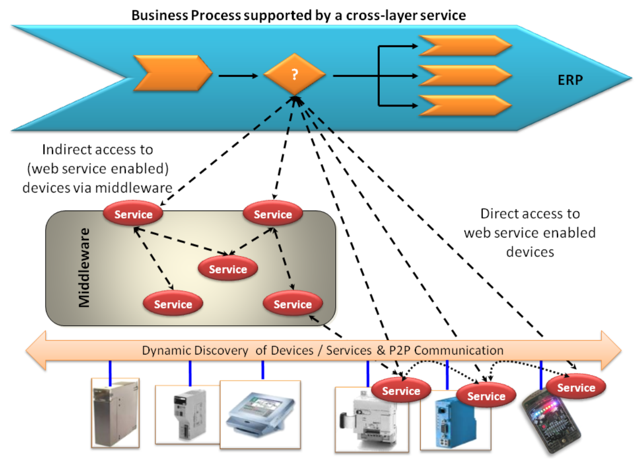


Fig. 1. A cross-layer web-service composition.

Integration of the devices on the functional level allows us to focus on orchestrating services based on their role in a process, and not the device as such. Collaboration is enhanced among the entities which leads to the minimization of islands of heterogeneous devices and boosts interoperability.

Service based integration of shop-floor devices with enterprise systems, brings many benefits in terms of business automation, response time, and data quality. Although these benefits make this integration highly desirable in a competitive economy, the unsupervised integration of devices with backend systems can also cause economic losses. These losses include: production halts, production time increase, reputation loss due to delays and even product recalls. When unexpected situations occur in the shop-floor, a rapid and dynamic adaptation of the business process is required in order to mitigate the effects that such an event can cause.

Hence, a beneficial integration of shop-floor devices with backend systems should provide characteristics that enable business processes to dynamically adapt to changes in the state of the device layer. With the current improvement of shop-floor devices and the adoption of SOA on all layers of the system, it is possible to create systems that are self-healing, self-monitoring, and self-optimizing.

Our aim is to depict how we can move towards highly-dynamic enterprises. Although simulations have been used before either at the shop-floor or at business process level, these have been used in an isolated way. Here we try to bring together the shop-floor and the business system and assess dynamically their state using also simulation and monitoring to adapt situations and predict possible problems. A SOA based middleware, such as one we proposed in [4] and demonstrated in [5], showed preliminary promising results in this area. Our **contribution** in this paper, is to extend our a SOA based system to accommodate simulation and analytics as well as decision making and process mapping strategies. This helps enterprise systems to dynamically adapt to changes in the shop-floor, to reduce the gap between the real world and its digital representation, and also to optimize business processes.

## II. State-of-the-art

Simulating Shop Floor production models and analyzing it with expert systems was done in the late 80's using Fortran and Prolog [6]. A scheduling system that optimized work flows based on shop floor and the ERP data and allowing the user to enter manually non-system data was proposed [7]. Another web based configuration and simulation tool for production systems [8] concerned only the conceptual phase of new production systems. Complex production planning and scheduling problems have been dealt based on a architectural approach [9]. Other tools like SIMSCRIPT-II.5 and SLAM System have been used in modeling work flow processes [10]. A simulation model based on data from the ERP system including a simple scheduling logic was developed [11]. The WITNESS modeling software is widely accepted in the industry for simulating production models, e.g. in production scenarios and analysis of best practices in production modeling [12], [13]. Autonomous control of production systems on the shop floor has been treated as discrete-event and continuous flow models in a simulation environment [14]. Recent efforts in the COLL-PLEXITY (www.coll-plexity.com) project focus towards a generic model of complexity. Such a model is considered to be a problem-to-system match framework for collaborative systems on the shop floor [15]. Supply Chain networks are modeled to understand the future demand on customized mass products [16]. Modeling the job flow, comparing the real time data to the models and enabling an expert system to make a decision for the enterprise application using service oriented paradigms is rather new. Furthermore the approach taken in this paper is holistic and tries to strongly couple the shop-floor and enterprise systems, using among also tools for simulation and estimation.

## III. Towards a more dynamic and autonomous system

As we move towards the "Internet of Things" [17], it can be expected that billions of devices of different size and capability

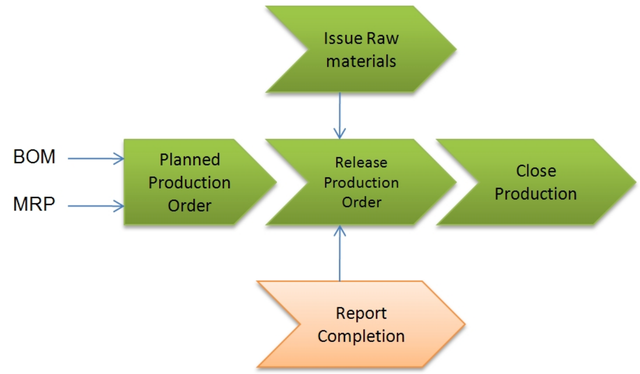


Fig. 2. Example business process for manufacturing production

will be connected and interact with each other and with business systems over IP (Internet Protocol) [18]. Generally for modern enterprises we expect [19]:

- Increase in complexity of systems.
- Increase in the heterogeneity of devices, software platforms, online services, etc.
- Wireless connection of a large proportion of devices to the backbone infrastructure.
- Increase of bandwidth and computing power.
- Ad-hoc computing, collaboration, task delegation, and environmental adaptation will be basic necessities.
- Vital on-demand software and service deployment.
- Gain in the importance of security and its satellite services.

In such future infrastructure, autonomic systems are expected to be of considerable help, since they are able to be, at a great degree, self-sustained and also to react to a dynamically changing environment. The essence of such autonomic systems is to deliver an optimized system. Four functional areas are defined: self-configuration, self-healing, self-optimization, and self-protection.

There are two strategies in achieving autonomic behavior, i.e., through adaptive learning and via integral engineering into systems [20]. Our approach focuses mainly on how to engineer such an autonomous system with respect to the coupling of business systems and real world production facilities, while adaptive learning, or self-learning, used at specific places e.g. identification of machine state.

## IV. Scenario

The short lifecycle of manufactured products (e.g. mobile phones, clothing, features in cars, etc), combined with their high amount of complex features, imposes great challenges to modern production systems. Due to their short lifecycle, assembly lines need to be flexible in order to easily adapt to new products. In this dynamic environment, the reduction of manufacturing costs drives the innovation on production systems.

In Figure 2 a generic business process for manufacturing a product is presented. While planning for production, the

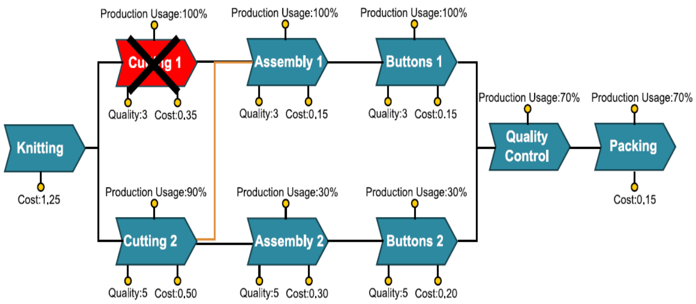


Fig. 3. Self-Healing and self-optimizing process re-route

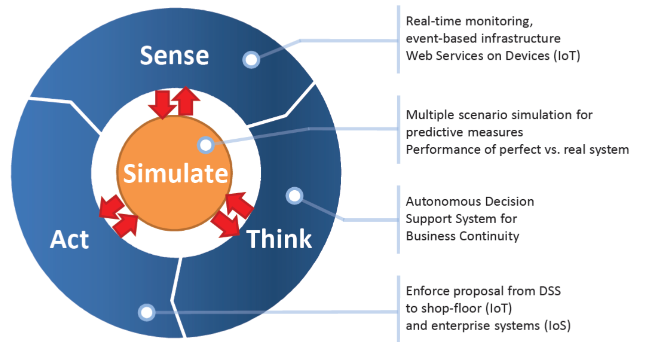


Fig. 4. Towards autonomous systems via cross-layer monitoring, simulation and management

bill of materials is identified. Information is taken from the audit where raw material availability is retrieved from the material master database. Once the production is planned, it is released. A Manufacturing Execution System (MES) takes care of retrieving this information and executing it on the machines. It keeps the ERP system updated about the current status of the production process.

Traditionally, these systems are huge databases and real time processing systems coming from different IT companies. Making information transparent to these systems on the IT landscape is a tremendous customization effort. Service Oriented Architectures make seamless integration to these systems possible and this in turn helps a MES system to connect directly to a supply chain system transaction without hassle and at the same time delivering the status of the production, to a production monitoring system. But when it comes down to devices on the shop floor, these are not visible to the ERP world.

When unexpected events in the production line occur, the ERP system need to investigate possible collateral effects, such as: order delays, maintenance, costs etc. Therefore, it is important to increase the integration between the shop-floor and ERP systems, in order to improve the decision making process and identify effects of unexpected situations in early stages.

### A. Production Halt

In Figure 3, a simplified T-shirt production line process is presented. In this production line, T-shirts are produced with different qualities, according to the production request. Each production step has a cost associated to it, which is dependent on the quality of the process. These costs usually vary with the number of items produced. For sake of simplicity, we assume in this scenario a fix cost for each process step.

In this production line shirts are produced, right from knitting, cutting, assembling with buttons, proving and packing. Breaks in the sequential operations would bring the rest of the production line to a halt. It would be inefficient to stop the entire line if a machine failed at the *Cutting 1* stage, while there are alternative routes available.

For a dynamic reaction to a breakdown state, it is necessary to have online information of other machines. Automatic production re-routing is still mostly solved locally, while the ERP system is not informed, or informed with significant delays. This is due to the fact that an assembly machine (the stitching machine in our example) is purchased from a different manufacturer than the knitting machine. The machines use different protocols and have different computing speeds. An even more complex situation can occur if break down states exist in the assembly lines, and due to lack of information, a plant manager releases a production order in the ERP system for 1000 pieces of a knitted shirt to be delivered in a short timeframe.

The production orders stocks up in the ERP system, and the MES system have no information of the error in the production line. If shop-floor machines host web services, the interaction with the ERP system increases and such production crisis scenarios can be better predicted and managed.

## V. Integrating Business and Shop Floor Processes

To improve the automation of business processes executed in factories, we propose a reactive approach that follows the concepts depicted in Figure 4. A number of concepts and technologies should come together to ease information flow among the different components, predict behaviors and finally take actions based on careful analysis of the current state of the system.

The key technical concept to the whole approach is a cross-layer communication in order to provide effective monitoring, simulation and management. In this approach we target several characteristics of a autonomous system:

- Self-healing: Automatic discovery and correction of faults or possible preventive actions. The system can recover from well-known problems including those that can be dynamically identified based on the correlation of events (Complex Event Processing).
- Self-optimization: Automatic monitoring and control of resources of the system can be done, in order for the different components that recognize themselves in a goal-driven manner with respect to the environmental context they act on. In that case, early indicators can be correlated and emerging problems are easier to pinpoint.

Figure 5 depicts the architectural approach proposed in this paper. The core idea is the strong coupling of Enterprise

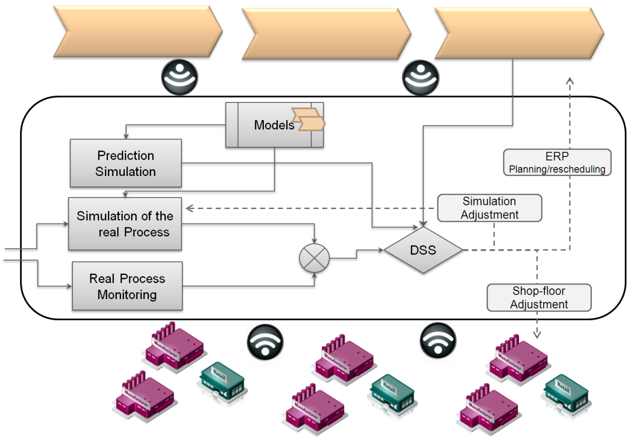


Fig. 5. Overview of the Reactive System

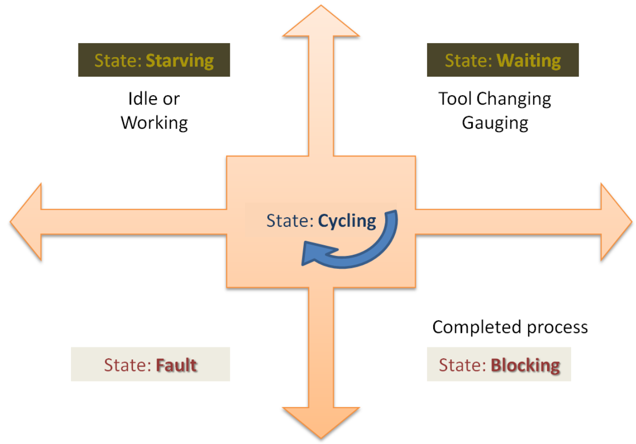


Fig. 6. Machine States

Systems with the shop-floor in a dynamic and autonomous way assisted by a Decision Support System (DSS). This DSS takes into consideration dynamic data coming from monitoring the shop floor, running simulations and its result are given as input to the business process control, the shop-floor and even on fine-tuning the simulation itself. In this concept, continuous real-time data flow into a monitoring system. In parallel a model of the shop-floor executes in a simulator. At specific intervals depending on the time or tasks, the output of the simulation and the monitoring are evaluated. Any deviation $\sigma$ is used as input from the DSS. Parallel to $\sigma$, the DSS considers inputs from the enterprise systems as well as the Prediction Simulation which predicts the next system state(s) of according to the existing models the system will continue to perform in the same way. The DSS considers all the input and makes decisions e.g. for optimizing the performance of the system, preventing faults that would happen if the mode of operation is unchanged etc. The DSS decisions are fed as input to the business systems, the shop-floor and the simulation itself, so that their behavior can be adapted. Having a precise information of the problems occurring (or predicted to occur) in the production line, the DSS can define measures to automatically modify the business process to heal the system. The result is that we are moving towards a self-* system that monitors and adapts itself according to the evaluation of the input from the sources mentioned.

### A. Simulation

Simulating a process work flow as a production model on a computer, takes away the risks of heavy investment. Modeling tools, such as flowcharts, process mapping, and spreadsheets are often used to identify how the shop floor would look like for a particular business objective. However such tools only show relationships between processes and generally don not provide any quantitative performance measures. They are static, deterministic and do now consider the dynamics of real life work in progress.

Dynamic production model simulation tools like WITNESS [21] and ProModel (www.promodel.com) consider the dynamic characteristics of production, e.g. process flow, processing times, setup requirement, labor, control rules, breakdown, shift, loading schedule, etc. We propose to go one step further by using the simulation results and comparing them to the live input coming the shop floor in order to assess the situation, possibly proactively determine problem zones, and optimize the shop-floor.

Figure 6 presents the generic states of a machine in a production line. Based on a known machine state a deterministic action can be performed. Nevertheless, defining the current state of the machine can be a challenging task. The information available to the back-end system consists of a great amount of events triggered through the assembly process. They could be identical to the simulated set of events and states, or deviate from the simulated conditions. Therefore, it is necessary to process these events in order to identify patterns that indicate which is the current state of the production line.

The advantage of production model simulation tools is that by changing the characteristics of production, the results on the shop floor can be accurately determined [22]. These tools give a good in depth understanding of how the manufacturing process would react to different situations on the shop floor. Such results can be considered as a reference for a series of deterministic characteristics of the shop floor [21]. As depicted in Figure 5, we propose a methodology where the workflow of a particular production process is continually monitored and compared to (pre)simulated results.

The state of the actual workflow process is continually monitored and compared with the simulated result on the shop floor. Based on the possible deviation $\sigma$, the work flow process or the business process can be affected. The deviations of the shop floor behavior from the expected result of the simulation have to be categorized in order to determine if the current condition would be tolerable or it would lead to a critical bottleneck. Various algorithms and methodologies can be combined to identify and categorize the state of the current production line.

## B. Self-healing Mechanisms

Through a comparison between the expected state (simulation results) and the real state of the shop floor, it is possible to identify malfunctions in the system. This information is essential for the system to self-heal.

For instance, in the example described in Section IV-A, if the machine responsible for executing the process "*Cutting 1*" fails, the production order of the T-Shirts with quality 3 will halt. This can result in delays, reputation loss, and even break in a contract, which implies high costs to the factory.

If the manufacturer defines a maximum cost for the T-Shirt production, it is possible to re-map the business process to avoid a production halt considering the new state of the system. This process re-mapping is depicted in Figure 3.

With the modification in the business process, the system self-heals and continues the production, while a maintenance workflow is triggered. Although the item cost increases, it remains within the threshold specified by the manufacturer, and prevents major losses due to production halts.

Another self-healing mechanism, investigated by this project, explores predictive maintenance and possible production bottlenecks. Based on real-time data from the shop-floor and having identified the current status of the production line, it is possible to predict the course of the current production. This is done based on analysis of the previous production history and also based on the simulated production model. The result of such prediction model analysis is forwarded to a Decision Support System (DSS), which then reacts providing input to the business process modeler (Figure 5). Finally, the integration of ERP business processes and such DSS can be performed through the SAP Manufacturing Intelligence and Integration (SAP MII) tool.

## C. Self-optimizing Mechanisms

Business processes are available as services in the enterprise service repository. Hence, a set of rules can be modeled in the DSS to invoke a corresponding business process at the prediction of a critical bottleneck state on the shop floor. Alternatively, shop floor devices hosting web services can be effectively used in such scenarios to prevent malfunction or break down of the machine. The DSS can reduce the production cycle on a particular assembly line when the system foresees a non linear increase in temperature or a production variable on the work flow.

In this article, we propose to base the optimization process on Swarm-Intelligent (SI) principles [23]. These methods were originally inspired by observation of various natural phenomena, in particular the collective behavior of social insects and flocking and schooling in vertebrates. The application of SI to distributed, real-time, embedded systems aims at developing robust task-solving methodologies by minimizing the complexity (including the intelligence) of the individual units (in our case machines of the assembly line) and emphasizing parallelism, and self-organization. From an engineering standpoint, the principal advantages of swarm-intelligent system design are four-fold: scalability, from a few to thousands of units; flexibility, as units can be dynamically added or removed without explicit reorganization; robustness, not only through unit redundancy but also through an adequate balance between explorative and exploitative behavior of the system, and simplicity (and low-cost) at the individual level, which also increases robustness. These properties would be highly beneficial if applied to machine production lines, and could be further optimized when machine have access to global information about the whole manufacturing process.

In particular, we propose to use Threshold-Based Algorithms (TBA) for a flexible task allocation mechanism to decide of the dynamic path in the production line. TBA have been initially used to model the dynamic task allocation decision process in ant colonies, and has been successfully applied for example for power-aware optimized load balancing [24]. Using TBA at the production-line level enables a reactive and fully-decentralized decision process done dynamically by the machines at runtime, based both on the objects to process, external data (environmental data, priority of the tasks, market values, etc), and the proprioceptive data of the machines. Threshold-based algorithms model group behavior based on a small number of control parameters (thresholds) that affect whether or not a particular task will be executed by a given machine. For this, every machine has an internal threshold value which is a function of different dynamic and static factors (price and time associate with task execution, machine current state, etc). Each task to process will have its own stimuli value that will be compared with the threshold of the machines and will be used to decide which machine will perform the task. Thresholds are allowed to change and become heterogeneous over time as a function of stimuli encountered and tasks performed, and this can lead to specialization and division of labor.

# VI. Conclusions

Shop-floor processes are more dynamic in nature than business processes. As service-enabled devices are evolving towards hosting web services, they provide seamless integration to MES systems rather and data coming from them can be effectively used in production modeling. This helps analyzing shop floor behaviors with low risk and investment. We proposed a system that can take the simulation of the shop floor behavior as a reference to compare with real time production data. We enhanced the proposal with a predictive model that can determine the course of the production line. It can be used together with an expert system to identify and categorize the states of the production line. The system can understand the shop floor, and business processes can react dynamically or even proactively to the changes in the production model. Industry standard tools like WITNESS and SAP MII are used to realize predictive maintenance scenarios, however they can be further extended to include simulation models of the shop floor and analyze real time data to estimate the production course.

## ACKNOWLEDGMENT

The authors would like to thank the European Commission and the partners of the European IST FP6 project "Service-Oriented Cross-layer infRAstructure for Distributed smart Embedded devices" (SOCRADES - www.socrades.eu), for their support.


## REFERENCES


[1] B. Automation. (2008, May) Wsd: Plug-and-play for building automation. Internet. Beckhoff GmbH. [Online]. Available: http://www.beckhoff.de/english.asp?press/news0408.htm

[2] S. Chan, D. Conti, C. Kaler, T. Kuehnel, A. Regnier, B. Roe, and D. Sather. (2006, Feb) Devices profile for web services. internet. [Online]. Available: http://schemas.xmlsoap.org/ws/2006/02/devprof

[3] O. Foundation. (2008, December) Opc unified architecture (opc-ua). Internet. OPC Foundation. [Online]. Available: http://www.opcfoundation.org/UA

[4] S. Karnouskos, O. Baecker, L. M. Sá de Souza, and P. Spiess, "Integration of soa-ready networked embedded devices in enterprise systems via a cross-layered web service infrastructure," in *2th IEEE Conference on Emerging Technologies and Factory Automation*, September 25-28, 2007, Patras, Greece.

[5] L. M. S. de Souza, P. Spiess, D. Guinard, M. Koehler, S. Karnouskos, and D. Savio, "Socrades: A web service based shop floor integration infrastructure," in *Internet of Things 2008 Conference, Zurich, Switzerland*, March 26-28, 2008.

[6] A. Flitman and R. Hurrion, "Linking discrete-event simulation models with expert systems," in *Journal of the Operational Research Society*, vol. 33, no. 8, 1987, pp. 723–734.

[7] A. Sivakumar, "Optimization of cycle time and utilization in semiconductor test manufacturing using simulation-based, online near-real-time scheduling system," *Proc. Winter Simulation Conf.*, pp. 727–735, 1999.

[8] W. Sihn, T. D. Graupner, T. Kuhlmann, and H. Richter, "Internetbasierte konfiguration und simulation von produktionssystemen," *Simulation und Visualisierung*, pp. 225–235, 2002.

[9] A. Artiba and E. Aghezzaf, "An architecture of a multi-model system for planning and scheduling," in *International Journal of Computer Integrated Manufacturing*, vol. 10, no. 5, 1997, pp. 380–393.

[10] C. Standridge and D. Steward, "Using expert systems for simulation modeling of patient scheduling," *Simulation*, vol. 75, no. 3, pp. 148–156, Sept. 2000.

[11] G. Weigert, S. Werner, and M. Kellner, "Fertigungsplanung durch prozessbegleitende simulation," *ASIM-Fachtagung*, vol. 10, pp. 42–51, 2002.

[12] S. Robinson, *Simulation: The Practice of Model Development and Use*. Wiley-Blackwell, 2003.

[13] R. Ahmed and S. Robinson, "Simulation in business and industry: How simulation context can affect simulation practice," *Spring Simulation MultiConference*, pp. 152–159, 2007.

[14] D. Armbruster, C. de Beer, M. Freitag, T. Jagalski, and C. Ringhofer, "Autonomous control of production networks using a pheromone approach," *Physica A: Statistical Mechanics and its Applications*, vol. 363, no. 1, pp. 104–114, Apr. 2006.

[15] G. Schuh, L. Monostori, B. C. Csjib, and S. Dringa, "Complexity-based modeling of reconfigurable collaborations in production industry," in *CIRP Annals - Manufacturing Technology*, vol. 57, no. 1. Elsevier Ltd, 2008, pp. 445–450.

[16] J. Vncza, P. Egria, and L. Monostori, "A coordination mechanism for rolling horizon planning in supply networks," *CIRP Annals - Manufacturing Technology*, vol. 57, no. 1, pp. 455–458, 2008.

[17] E. Fleisch and F. Mattern, *Das Internet der Dinge: Ubiquitous Computing und RFID in der Praxis: Visionen, Technologien, Anwendungen, Handlungsanleitungen*. Springer, 2005.

[18] J. W. Hui and D. W. Culler, "Extending IP to Low-Power, wireless personal area networks," *Internet Computing, IEEE*, vol. 12, no. 4, pp. 37 – 45, July 2008.

[19] S. Karnouskos, "Autonomic sensor networks for future enterprise services," *KiVS 2007 Conference*, vol. 1, pp. 19–21, 2007.

[20] R. Sterritt, M. Parashar, H. Tianfield, and R. Unland, "A concise introduction to autonomic computing. advanced engineering informatics," *Autonomic Computing*, vol. Vol 19, Issue 3, pp. 181–187, 2005.

[21] R. Ahmed, T. Hall, P. Wernick, S. Robinson, and M. Shah, "Software process simulation modelling: A survey of practice," in *Journal of Simulation*, vol. 2, no. 2, 2008, pp. 91–102.

[22] M. Pidd and S. Robinson, "Organising insights into simulation practice," *Proc. 2007 Winter Simulation Conference*, pp. 771–775, Dec 2007.

[23] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence From Natural to Artificial Systems*. Oxford Univ. Press, New York, 1999.

[24] C. Cianci, V. Trifa, and A. Martinoli, "Threshold-based algorithms for power-aware load balancing in sensor networks," in *Proceedings of 2005 IEEE Swarm Intelligence Symposium, IEEE-SIS*, Passadena, CA, USA, 2005.