2010 International Conference on Complex, Intelligent and Software Intensive Systems

A Pattern-Based Coordination and Test Framework for Multi-Agent Simulation of Production Automation Systems

Thomas Moser[1] Munir Merdan[2] Stefan Biffl[1]
[1]Christian Doppler Laboratory for Software Engineering Integration for Flexible Automation Systems, Austria
{thomas.moser, stefan.biffl}@tuwien.ac.at
[2]Automation Control Institute, Vienna University of Technology, Austria
merdan@acin.tuwien.ac.at

***Abstract*— Production automation systems consist of many entities (like robots and shuttles) that interact in complex ways to provide the overall system functionality like product assembly. Multi-Agent Systems (MAS) can simulate these system entities and their interactions to better understand the system behavior according to production strategies and system configuration, which are otherwise hard to determine before their (costly) implementation. A main challenge is to coordinate the groups of agents effectively to reach a common production goal, e.g., fast and robust delivery on customer orders for assembling a set of products. In this paper, we introduce the pattern-based extension of the MAS-based manufacturing agent simulation tool (MAST) to support the simulation and evaluation of business processes in an assembly workshop: a layered coordination and test framework has been implemented and evaluated in cooperation with industry partners. Domain experts found the coordination patterns usable and useful to elicit their knowledge. The test framework allowed exploring the effects of production strategies on the system 30 times faster than with a real-world hardware workshop.**

Keywords- Production Automation Simulation; Coordination in Multi-Agent Systems; Coordination Patterns.

I. INTRODUCTION

Modern production automation systems need to become more flexible to support the timely reaction to changing business and market needs. However, the overall behaviour of the many elements in a production automation system with distributed control can get hard to predict as these heterogeneous elements may interact in complex ways (e.g., timing of redundant fault-tolerant transport system and machine groups) [9].

Software agents seem particularly well suited to model and design flexible, modular, and self-organizing systems that are robust to changes in their environment. Multi-Agent Systems (MAS) can help simulate in a distributed control system the effects of production strategies that coordinate the behavior of the entities in the production automation system [5]. However, designing coordination for several levels of agents in a flexible control structure is a major challenge and can benefit from design patterns that can be systematically validated and reused in a range of contexts. In the production automation domain there are the levels of a) business processes; b) redundant robust transport system and machine functions; and c) control components in the machines of the production automation system (see Figure 4).

In order to allow the interaction between these levels, coordination between agents on each layer is required. Coordination patterns are design/implementation paradigms suitable for solving certain problem scenarios. In our context, coordination patterns are used to enhance the design and implementation of a) domain expert knowledge in their particular layer (model of world, constraints, strategies, goals, etc.); b) the allocation of available resources (e.g., production strategies, auctions, or work load balancing); and c) information model and control (e.g., master/slave (following the layer concept), message exchange, or blackboard pattern).

In this paper, we describe the pattern-based extension of a MAS-based production automation simulation tool, Manufacturing Agents Simulation Tool (MAST) [17]. The architecture of our extension, the MAST-SAW (Simulation of Assembly Workshops) framework consists of three major parts: a) the original MAST simulator; b) the work order scheduling system; and c) the performance test management system (PTMS). The work order scheduling system transforms incoming business orders into feasible working tasks and then coordinates these tasks with the original MAST simulator. The PTMS supports the creation and execution of test cases to measure system performance for exploring the systematic effects of a range of strategies and system configurations and to enable the quantitative evaluation of a large number of simulation runs. We shortly report on initial experiences with the MAST-SAW framework for simulating the existing pallet transfer research system located at --- (for more details refer to [11, 12]) with 120+ distributed machine and device controllers. The simulation system has been validated with the hardware of this research lab to ensure the external validity of simulation measurement and analysis.

The remainder of this paper is structured as follows: Section 2 summarizes related work on MAS, their application in production automation, and on coordination patterns; Section 3 motivates the research issues. Section 4 describes the coordination and test framework MAST-SAW and Section 5 shows the results of the statistical data analysis for a typical usage scenario of MAST-SAW. Section 6 discusses the findings and Section 7 concludes.

978-0-7695-3967-6/10 $26.00 © 2010 IEEE
DOI 10.1109/CISIS.2010.56

II. Related Work

This section summarizes related work on Multi-Agent systems, their applications in the production automation domain, and on coordination patterns.

A. Multi-Agent Systems

A Multi-Agent System (MAS) is a network including more than one problem solver. The solvers can interact with each other to find a solution to a given task which is beyond the individual knowledge and capability of each problem solver. The so-called problem solvers are represented as agents. [5]

MAS often represent real-world problems mapped into a software system. These problems are often very complex, distributed, and can change dynamically and frequently. Therefore systems dealing with such problems have to be modular and abstract. A MAS shows these abilities and fulfils them by having "a number of functionally specific and (nearly) modular components (agents) that are specialized at solving a particular problem aspect" [16].

An agent within a MAS is usually defined as autonomous, which means that it can operate without any intervention of any other component or human being and has its own control over its (local) actions and states. The agent has some kind of social ability, which means that it can communicate with humans or other components, and is reactive, so the agent can collect information about its environment and react autonomously on relevant events.

Lazanský et al. [7] define the following advantages of using MAS for production planning: a) clear separation of collaboration and problem solving knowledge; b) transparent organisation of knowledge and data enabling easy maintenance; and c) highly dynamic behaviour with minimal communication traffic achieved through the subscription mechanism. Besides these advantages and the effectiveness of MAS there are also some disadvantages and open issues to consider [16]: a) the agent-based problem solving mechanism is not always the optimal way and may result in computational instability, i.e., a useful solution may not be found in limited computational time; b) MAS can not be used for physical problems that can not be divided into sub-problems or sub-objectives; and c) agent-based systems often consume significant resources, especially for monitoring and support. Implementing, testing and modifying the system can take substantial effort.

B. MAS in Production Automation

The complexity of solutions in the manufacturing domain tends to increase due to new technological developments and market demands. Current manufacturing systems are mostly based on hierarchical or centralised structures and thus are often rigid and not robust and adaptable enough to react effectively to unpredictable disturbances. The multi-agent technology, has a decentralized control architecture that provides advantages such as: heterogeneity, modularity, increased flexibility and robustness against failures; thus this approach seems well suited to manage the dynamic nature of manufacturing environments [3].

Substantial research has been devoted to apply the MAS paradigm in a wide range of manufacturing fields: manufacturing control [17], integrative business information systems [6], process planning [18] and scheduling [15]. However, although confirmed as promising approach and deployed in a number of applications recently, widespread adoption of agent-based concepts by industry and governments is still lagging. The lack of awareness about the potential of agent technology, concerns regarding MAS stability, scalability and survivability as well as missing standards, design and development tools have been seen as major obstacles that have to be overcome in order to achieve progress in the adoption of agent technologies [13]. Simulation is considered as effective method for testing different control architectures and improving the quality of solutions, which are in many cases closely related to control systems and scenarios in real life manufacturing [14].

ProPlanT (Marík et al. [10]) is a multi-agent-based prototype for production planning, which is implemented to simulate production processes of TV transmitters manufactured by Tesla TV. The system consists of a number of agents that are classified into three groups. The first group includes so-called project planning agents (PPA) that are responsible for fulfilling one order. The second group consists of project management agents (PMA) which aim at "distributing the work to the best possible executive production agents (PA)" [7]. The third group includes PAs that are responsible to carry out the manufacturing processes at the shop floor level. ProPlanT has a similar agent structure as the MAST system on which the new MAST-SAW simulation is built.

Lim and Zhang [8] define an agent framework for an agent-based manufacturing system consisting of four manager agents, which can have a certain number of subagents. These agents can be classified as execution agents, who carry out procedures and make decisions, and as information agents, which provide information. The product manager agent (PMA) "represents all the physical products that are currently and previously manufactured in the shop floor" [8]. The resource manager agents (RMA) are physical resources like, machines or conveyor belts. Task manager agents (TMA) are the interface between agents and the user. The execution manager agent (EMA) is responsible to monitor the agents producing a certain product. This agent has subagents which have different roles and abilities. If changes within a product plan occur or a new product is added to the simulation the EMA sends a message to the process agent (PA). The PA has to check if all manufacturing processes necessary for production are available. The machine grouping agent is responsible to generate alternative machine groupings based on the process sequences provided by the PA. The scheduling agent (SA) provides a certain number of scheduling options and the optimisation agent (OA) evaluates and optimises the scheduling options. While these collection of agents can well support the simulation of a manufacturing system, there are very few reports on the validation of the framework, which is necessary to allow comprehensive evaluations of manufacturing system parameters before the actual implementation of the system in a real-world workshop.

C. Coordination Patterns

Coordination is the organization of a set of entities to achieve predictable system behavior, a coordination pattern within a MAS describes the communication and the negotiation between groups of agents. This information exchange can either occur between agents or between the agents and an outside role, like human beings or other applications.

A coordination pattern is a special kind of software design pattern, which has roots in the design of building architecture [2]. Coordination patterns always express the relation between a certain context, a problem, and solution options. The adoption of coordination patterns to the MAS domain promises to alleviate problems like lack of agreed definition, duplicated effort or common vocabulary [4]. The documentation of the most import building blocks of the system, which already have been accepted by the software community, provides other researchers with reusable and adaptable solutions to a given task. The abstract description of system parts allows understanding the principles of the pattern without very specific knowledge of the system or language features. While coordination patterns promise to enable the coordination among multiple levels in MAS systems, there are very few reports on full vertical evaluation of a MAS system in production automation, which span from top-level business processes to the simulation of hardware machines and devices.

III. Research Issues

Recent projects with industry partners from the production automation domain raised concerns about computer-based simulation of typical production automation processes, which need an abstract and adaptable representation that is easy to understand both for domain experts and for software engineers. In order to ensure an efficient and effective simulation of production automation processes, MAST seems to provide a suitable base for performing software simulation. However, a central challenge is how data structures and processes originating from different domains (e.g. business processes, sets of machines, communication networks, and physical devices) can be modeled in the MAST simulation software with reproducible and validated results. In order to address this challenge, we derived the following research issues.

RI A: Coordination between domain layers. Analyze the production automation sub-domains (business processes, workshop processes, and physical machines) to identify coordination patterns that encapsulate the expert knowledge (e.g., model of world, constraints, strategies, and goals) for flexible and robust design and implementation of coordination in and between these layers.

The coordination patterns used in this work describe the communication and negotiation between agents within the MAST simulator and agents within the work order scheduling component for the allocation of available resources (e.g., production strategies, auctions, or work load balancing). The patterns define a reusable guideline for researchers to develop an interface between the business layer and the job shop layer of an agent-based production planning system. These patterns facilitate consistent and verifiable design and implementation (e.g., master/slave (following the layer concept), message exchange, or blackboard pattern).

RI B: Performance measurement for quantitative evaluation of coordination patterns in the production automation system. The production automation system employs a considerable number of parameters to define the system structure and behavior on several layers. Thus a large number of simulation runs, typically hundreds or more, is necessary to allow gathering sufficient data to systematically explore and evaluate the impact of production strategies, configuration changes, or coordination approaches before deciding how to build the real system.

For investigating these research issues we gathered requirements with industry partners from the production automation domain. Then we designed and implemented a work order scheduling component and a performance test management component on top of the MAST software simulator to handle incoming business-level customer orders, the MAST-SAW. In order to evaluate the MAST-SAW framework we performed a quantitative performance evaluation with 1,000+ test-cases, measured key factors (like the average throughput, machine utilization rates) as foundation for systematic statistical data analysis and discussion of the results with domain experts. Compared to the MAST-SAW framework the agent coordination described in [8] is more comprehensive; but primarily aims at the intra-simulator coordination and at the coordination with a user, rather than at the coordination between different domains (e.g. coordination of workshop with business processes) as introduced in this work.

IV. COORDINATION/TEST FRAMEWORK

Coordination within MAS is one of the main issues to ensure that a community of agents acts in a coherent manner. Coherence means that the agents within a group follow a common goal and do not conflict with each another. Usually agents communicate with each other to solve a given task. They distribute their goals, intentions, results and states to their agent group. The coordination within the agent community in this paper is described in patterns which provide a reusable solution to similar problems.

A. Framework Architecture

The coordination of the agent-based production planning system is the main goal of the framework presented in this paper. The production planning system is based on a simulator developed by Rockwell Automation, which already includes a set of FIPA-compliant simulation agents [1]. This agent system is modified and extended by a coordination component that should monitor the agents' tasks and activities. Figure 1 provides an overview on the layers of the agent system: on the business layer the dispatcher (order agent) converts customer orders into work orders that are sent to the workshop layer (product agent).

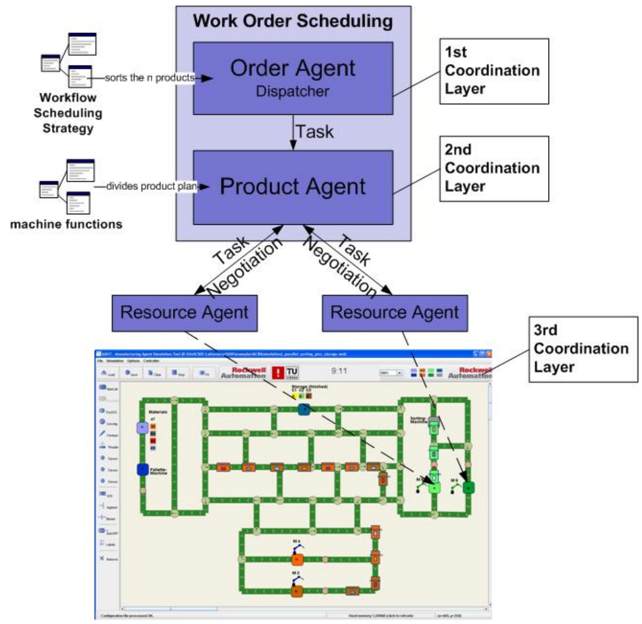

Figure 1: Coordination layer model of the framework.

Figure 1 shows the layer model of the production automation system used in the MAST-SAW framework.

Business process. The order agent in the first layer is responsible for the incoming business orders from the costumer, monitoring and guiding a single product through the simulator. The business orders represent guidelines for the arrangement of product sequences depending on the selected workshop scheduling strategy. After the sorting mechanism the dispatcher registers n agents (PA) for each product and forwards the product, as a defined so-called product plan, to these agents.

Workshop Scheduling. These n PAs operate in the second layer of the coordination model and represent the interface between the dispatcher and the simulation agents. They analyze the product and divide the product plan in more detailed working steps, i.e. transport and assembly steps. These working steps are delegated to Resource Agents (RAs). This decision making process includes a very important negotiation and allocation sequence between the PA and the RAs. The choice which RA will get the working steps follows an auction pattern. The Auction pattern provides the possibility of offering a good or a service to other participators. In this case the PA sends an announcement message, including an identifier and the machine function, to the RAs. The RAs offering the required function send back a message containing the estimated processing time of the machine function plus the estimated time needed for the transportation to the machine. The PA picks the RA with the lowest overall machine function time and delegates the task to it. During the simulation processes the product agents of the second layer can influence the simulator and monitor the events and states within the simulator.

MAST. The third layer is the simulator, where the simulation agents communicate with each other as well as with the agent of the layer above. The agents can be classified into Resource Agents, which work on tasks delegated by the PAs, and Transport Agents, which have to fulfil various transport steps (e.g. sending palettes). The simulation agents implemented by Rockwell Automation already include some self-coordinated behaviour such as flexible routing. This coordination functions have been modified and adjusted to the needed requirements. The agents have to fulfil the various tasks getting from the agents in the second layer and have to report about their status, capacities, failures and the measurements.

B. *Agent Negotiation Example*

The coordination component represents the interface between the business layer and the workshop layer. The coordination component is described in a Master/Slave pattern which is shown in Figure 2. The dispatcher (Order Agent), located in the first layer, acts as Master and delegates tasks to a certain number of Slaves depending on the number of products in the order. These Slaves are defined as Product Agents (PA). These agents in turn act as Masters and use the simulation agents (Resource Agents and Transport Agents) from the third layer as Slaves to fulfil their tasks.

The production planning process starts with the registration of the Order Agent (OA) which gets an order consisting of one or more products. The OA orders the set of products into a sequence defined by the workflow scheduling strategy and registers for each product one Product Agent (PA). The PA analyses the product and divides it into more detailed machine functions. Next the PA sends an announcement message, including the identifier and the needed machine function, to the Resource Agents (RA) and delegates the machine functions. The RAs, in turn, use the Transport Agents (shuttle, conveyor belts, crossings and stopper) to send products from one point of the simulator to another. As soon as a (sub-)product arrives at its destination, the RAs inform the PA responsible for the product. The PA in turn analyses its product plan and checks if further production tasks have to be fulfilled. If this is the case the PA sends again a message to the RAs asking for the service needed. If all production steps are fulfilled and the end product arrived at its destination the PA informs the OA about the state and deregisters itself.

C. *Examples for the Implementation of Coordination Patterns in MAST-SAW*

In software engineering a pattern is a common vocabulary to describe a general solution to a design problem. It is a prototype for software solutions, describing the most important parts of a system in an abstract way. They should help researchers to resolve recurring problems based on past experiences. This section exemplarily describes two patters used in the MAST-SAW framework.

Master/Slave Pattern. The Master/Slave pattern describes the hierarchical distribution of defined tasks. The pattern increases the reliability, performance and accuracy in the system. Besides parallel working is possible because during the Slaves fulfil the tasks of the Master, the Master can work parallel on other problems as well. In this project the production processes are monitored by a so called

Product Agent (PA), which gets the product plan of one product. It represents the Master who communicates with the simulation agents and distributes tasks to the Slaves. The Slaves in the project are the simulation agents, like shuttles, machines or belts. The problem to be solved is the product plan which has to be fulfilled. The Slaves can also act as Masters. They may have Slaves which fulfil their tasks. In the project there are four layers of Master/Slaves relationships as seen in Figure 2.

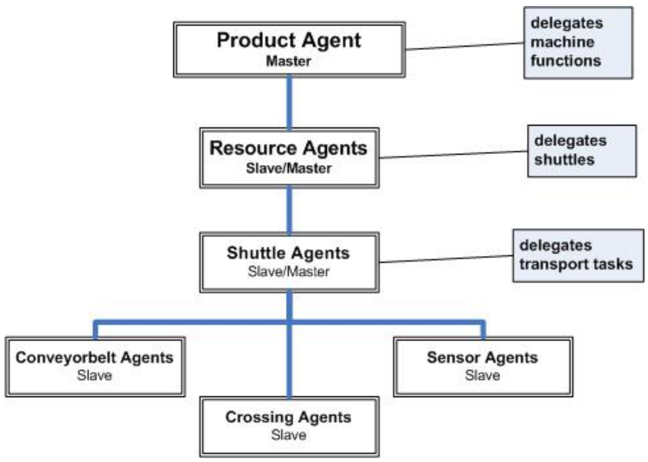

Figure 2: Master/Slaver levels in the MAST-SAW framework.

Blackboard Pattern. The Blackboard pattern provides a centralized messaging/data mechanism. A Blackboard is used to hold data; meanwhile other agents can subscribe to get access to the data of their interest. In the project agents have to solve a production problem, which involves transporting (sub-)products to different destinations (machines). During the problem solving process the agents have to collaborate with each other. The coordination of the individuals, which have various abilities and tasks to complete, is realized by a messaging system. The agents send messages as soon as their state changes or an event is deployed. A subscription function allows the agents to receive just the information which influences their behaviour.

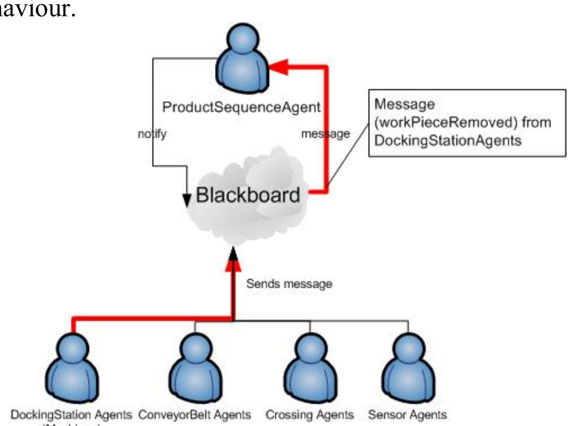

Figure 3: Blackboard Pattern in the MAST-SAW framework

For example, as shown in Figure 3, the Product Agent (PA) is responsible for the production of a certain product. Therefore it has to communicate with the agents system and monitor if the simulation agents have fulfilled all the tasks it has delegated to them. During the assembly process the simulation agents send messages to other agents to publicize their states and actions. The PA is just interested in messages of resource agents (machine) because it just needs to know when a (sub-)product is removed from the simulation. Therefore it is subscribed to machine agent messages.

D. *Performance Test Management System*

Figure 4 illustrates the design of the performance test system that allows automatically running a large number of systematic variations of test scenarios to evaluate the effects of these variations on the overall system performance. The test system is a harness that starts up the MAST-SAW with the current test scenario parameters and customer order events. The MAST-SAW in turn runs the software agents, the coordination patterns, and logs the results of the test run. The test system collects the measurement results of each test run in a database for further statistical analysis.

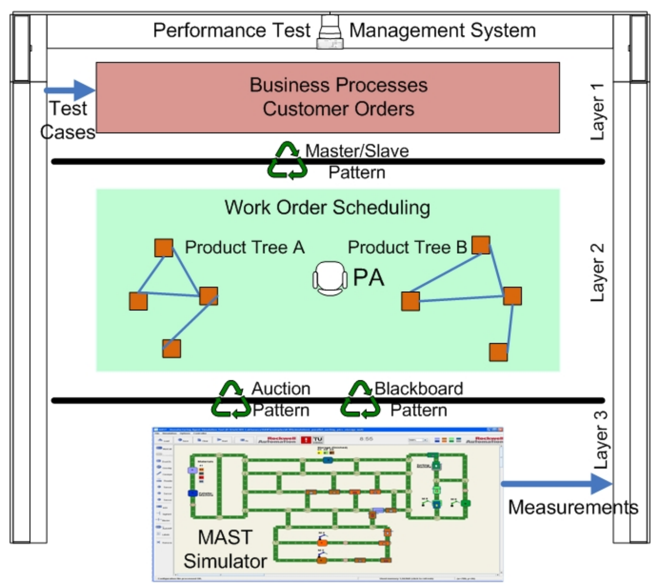

Figure 4: Performance test system in MAST-SAW.

The *Performance Test Management System* (PTMS) is a part of the MAST-SAW framework and uses the MAST system for automatically running pre-defined sets of test cases described in XML files. These test cases describe goals, strategies, and constraints: the products to assemble, assembly steps, the workflow scheduling strategy to apply, the number of pallets to use, and the duration of a production shift. For running a test suite, the MAST system is reset to a starting state, the XML file is parsed and the test cases are consecutively injected into the MAST system, which acts on the input parameters of the test cases to run the agent-based simulation and control system. Relevant events and result data (e.g., number of finished products, machine utilization rates) are measured using pluggable measuring methods and algorithms to the agents and then logged to an XML output file. This approach allows scheduling automated runs of a large number of systematic test case variations, resulting in comprehensive output data for statistical data analysis. In addition to the automated runs of test cases, the PTMS provides a generator for preparing a given number of systematically derived test cases, e.g., for evaluating the impact of the variation of one or more parameters in the simulation (e.g., with exhaustive enumeration of the parameter range or with statistical sampling).

V. QUANTITATIVE EVALUATION STUDY

To illustrate the improved level of capabilities for quantitative empirically evaluation possible with the new MAST-SAW framework, we identify two scenarios to measure the performance of the system and we conducted 1,000+ performance test cases as input to the MAST-SAW framework.

A. *Production Strategies, Pallets and Failures*

The first scenario is to measure the changes of throughput of the system (number of finished products) when we add more pallets into the workshop floor. This scenario is also useful to assess how well the agents coordination when the number of transport agents (i.e., pallets) is increased which add the complexity of the coordination. The second scenario deals with different types of failures introduced into the system during run time. Here we once again measure the system throughput when it has to face failures with different severity level. The measurement of throughput can be used to assess the capability of the agents to work on failover strategy when they have to face a failure case.

Table 1: Some Workshop Scheduling Strategies.

Strategy	Abbr.	Description
First Come First Served	FCFS	The first allocated task is executed first.
Earliest Due Date	EDD	Gives the highest priority to a task with the earliest due date (derived from the work order due date).
Shortest Processing Time	SPT	The task with the shortest processing time is sequenced first.
Critical Ratio	CR	Is the time remaining till due date divided by the work remaining. A task with a lower CR is given higher priority than tasks with higher CR.

A total of 1,085 test cases were generated from the scheduling strategies as input to the MAST-SAW. The test cases were split into 7 batches which were run in parallel on a high-performance test server. The overall performance test runs took about 26 hours per batch (a test run on the lab hardware would take on average around 30 times longer than the software simulation as the simulation can run faster than in real time and in parallel instances). Later we collected the results for statistical data analysis. Each test case consists of following parameters:

1) a workflow scheduling strategy (see Table 1), please refer to [12] for detailed description of the selected scheduling strategies.

2) the number of operating pallets during the production (10, 15 or 20).

3) failure specifications (refer to Table 2 for more details). The failure specification consists of the identifier of the affected resource to fail, the start and end points in time of the occurrence of the failure. We classified the risk of a failing conveyor (according to the position and the importance of the conveyor for the overall system) for all conveyors in the workshop in 5 failure classes. For effective comparison of the robustness of workflow scheduling strategies regarding their exposure to failures in the transportation system, failures with the same specification were used for all workflow scheduling strategies

4) a workload of 25 orders. An order consists of a product type to be built and a randomly generated due date for the product.

5) the shift (production) time for each test case was limited to ensure that 25 randomly selected orders could not easily be finished without a proper scheduling strategy and agents' coordination.

Table 2: Failure Classes.

Failure Class	Failure Impact
F0	No failure
F1	Failures of redundant conveyors which cause almost no detours
F2	Failures of redundant conveyors which cause long detours
F3	Failures of conveyors resulting in the unreachability of a single redundant machine
F4	Failures of conveyors resulting in the unreachability of multiple machines

B. *Data Analysis Results*

We use descriptive statistic and analyze the results of each production run. The results show that adding more pallets will increase the system throughput (number of finished products) for all scheduling strategies. When using 10 pallets, Shortest Processing Time (SP) offers the best overall production performance, however Critical Ratio (CR) has the best performance when increasing the number of pallets from 10 to 20. In the second scenario, we analyze the results of introducing different transportation failure severities into the simulation.

Figure 5 outlines that for all strategies introducing a failure will reduce the average system throughput compared to production without failure. In overall CR offers the best performance for handling all type of transportation failures compared to others strategies. Our descriptive analysis reports that by adding the F1 reduced the average number of finished products depending on the strategy such as: CR (-5%), EDD (-11%), FCFS (-6%), and SP (-1%). Similar patterns of decreased system throughput also can be seen when we introduced the remaining of the failure classes into production. Here we concluded that the combination of CR with 20 pallets will provide the best performance of the system and CR has the best capability for dealing with transportation failures in our study context. For a more detailed examination of the data analysis including statistical hypothesis test results please refer to [11, 12].

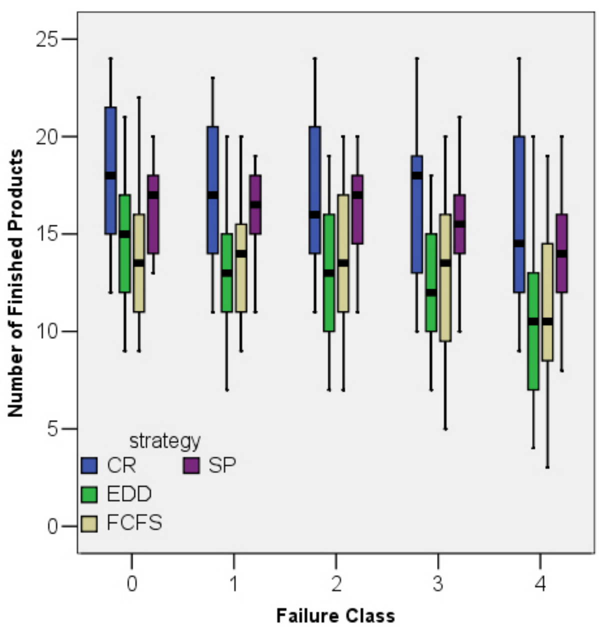

Figure 5: System throughput for 5 classes of failures.

VI. DISCUSSION

This section discusses the findings, benefits, and limitations of this work regarding the research issues. Additionally, the extensibility and fault-tolerance of the MAS architecture is described.

RI A: Coordination between domain layers. The production automation simulator is structured using a three-layered architecture. As described in section IV.A, the first layer is responsible for handling the incoming business orders. The second layer is in charge of handling the production of a single product and deriving more detailed working steps for this production. Finally, the third layer represents the existing software simulator and is responsible for managing the single tasks derived in the previous layer.

As shown in section IV.A, coordination is applied to the MAS production automation simulator by means of adding a coordination component to the original system. The coordination within the existing simulator is organized by message exchange between the agents. The research for this paper supports that MAS can be an effective solution for modular, decentralized, complex and time varying systems, like production planning systems if they are validated with the hardware and supported with coordination patterns that facilitate the encapsulation of domain expertise. The complex production planning and modelling techniques can be simulated by agents, which negotiate and collaborate with each other to solve given tasks. This coordination within the agent community is similar to the situation within a factory where employees interact with each other or with machines. These employees and machines can be substituted by agents, which have different abilities defined in their behaviours. The information used to produce certain products is divided into various working steps and delegated to the agents.

A limitation that occurs during the implementation phase of this work is the huge number of messages sent by the agents. For each changed event or state a message is sent and received by several agents. This message overload is a performance problem and can be improved by analyzing the messages and filtering the ones which really affect the system. During the implementation phase of this work the easy extensibility of MAS was one major advantage for further extensions of the application. The agent system was extended by new agents and components without much effort. The new system parts were easily added and integrated. An agent-based application allows working with incremental software development processes which means that the system can be used without finishing all requirements.

RI B: Performance measurement for quantitative evaluation of coordination patterns in the production automation system. One aim of this paper was to explore and evaluate the performance of predefined workflow scheduling strategies on a digital simulator. Thus we conducted a systematic quantitative study with a large number of test cases. This evaluation acts also as a proof of concept, which demonstrates that the implemented coordination component is able to coordinate a complex and distributed agent-based system, while helping to compare the various workshop scheduling strategies with each other. The analyses show that the workflow scheduling strategies and configuration parameters have significant impact on the production system effectiveness. The first part of the analysis focused on the number of finished products in a certain shift, depending on the selected strategy and the number of pallets on the simulation. The second part of the analysis compared the robustness of the particular workflow scheduling strategies regarding transportation errors.

The simulation runs showed that the MAS already provided some fault tolerance. The coordination component implemented during this work provides a monitoring function represented by the dispatcher. If this component fails or the communication between the simulation agents and the coordination agents breaks down, the agents within the simulation still fulfill their tasks, but act less efficient as a group. The simulation agents can work without the coordination agents for a certain period, which allows the recovery from the failure to improve efficiency on the system level.

VII. CONCLUSION AND FURTHER WORK

In this paper, we introduced the pattern-based MAST-SAW multi-agent production automation simulation tool. We showed its usefulness for the simulation of complex and distributed production processes using a layered coordination framework. Further, we performed test run series in the context of a use case from the production and logistics domain to explore the systematic effects of production strategies and system configuration parameters to statistically evaluate the results. In order to ensure an efficient and effectively simulation of typical production automation processes, we derived and addressed the following issues.

Coordination between domain layers. The production automation simulator is structured using a three-layered architecture, with the first layer being responsible for handling the incoming business orders, the second layer being responsible for handling the production of a single product, and the third layer representing the existing software simulator. This layered architecture follows expertise and communication in the application domain and supports the elicitation of expertise and validation by domain experts; the case study has provided reasonable evidence that the architecture seems to be an efficient and effective way of integrating different domains into a MAS simulator. Domain experts found the coordination patterns usable and useful to elicit and validate their knowledge for requirements engineering and the overall system design.

Performance measurement for quantitative evaluation of coordination patterns in the production automation system. The MAST software simulator was calibrated with the respective elements in a hardware lab to support this aspect of external validity of the simulation. We collected empirical evidence by conducting a set of systematic test cases where the agents have to fulfil a certain set of tasks under different conditions. This demonstrates that the implemented coordination component is able to coordinate a complex and distributed agent-based system and additionally helps to explore and evaluate the various workshop scheduling strategies with each other. The test framework allowed to explore the effects of production strategies on the system 30 times faster than with a real-world hardware workshop as the software simulator could be put the system in start states immediately, run processes faster, and run several parallel instances.

Further research will include the introduction and evaluation of more complex workshop layouts, and the introduction of new machine failure scenarios for simulation. In addition, a set of more sophisticated workflow scheduling strategies will be implemented in order to simulate and analyze the benefits and limitations of dynamic dispatching.

ACKNOWLEDGMENT

The authors would like to acknowledge the works of the *Rockwell Automation Research Center*, Czech Republic, in the field of the Manufacturing Agent Simulation Tool (MAST), as well as *Clemens Gondowidjaja*, *Christoph Gritschenberger*, and *Alexander Pucher* for their support in implementing the different production strategies and measurements. This work has been supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria. In addition, this work has been partially funded by the Vienna University of Technology, in the *Complex Systems Design and Engineering Lab*.

REFERENCES

[1] "FIPA: Foundation for Intelligent Physical Agents," 2006.

[2] C. Alexander, *The timeless way of building*, Oxford Press, 1979.

[3] S. Bussmann, N. Jennings, and M.J. Wooldridge, *Multiagent Systems for Manufacturing Control: A Design Methodology*, Springer, 2004.

[4] D. Deugo, M. Weiss, and E. Kendall, "Reusable Patterns for Agent Coordination," *Coordination of Internet Agents: Models, Technologies, and Applications*, Springer, 2001, pp. 347–368.

[5] N.R. Jennings, and M.J. Wooldridge, *Agent technology: foundations, applications, and markets*, Springer-Verlag, 1998.

[6] R. Kishore, H. Zhang, and R. Ramesh, "Enterprise integration using the agent paradigm: foundations of MA-based integrative information systems," *Decision Support Systems*, vol. 42, no. 1, 2006, pp. 48-78.

[7] J. Lažanský, O. Štepánková, V. Mari´k, and M. Pechoucek, "Application of the multi-agent approach in production planning and modelling," *Engineering Applications of Artificial Intelligence*, vol. 14, no. 3, 2001, pp. 369-376.

[8] M.K. Lim, and Z. Zhang, "A multi-agent based manufacturing control strategy for responsive manufacturing," *Journal of Materials Processing Tech.*, vol. 139, no. 1-3, 2003, pp. 379-384.

[9] A. Lüder, J. Peschke, T. Sauter, S. Deter, and D. Diep, "Distributed intelligence for plant automation based on multi-agent systems: the PABADIS approach," *Production Planning and Control*, vol. 15, no. 2, 2004, pp. 201-212.

[10] V. Marik, M. Pechoucek, O. Stepankova, and J. Lazansky, "ProPlanT: Multi-Agent System for Production Planning," *Applied Artificial Intelligence*, vol. 14, no. 7, 2000, pp. 727-762.

[11] M. Merdan, T. Moser, D. Wahyudin, and S. Biffl, "Performance Evaluation of Workflow Scheduling Strategies Considering Transportation Times and Conveyor Failures," *IEEE International Conference on Industrial Engineering and Engineering Management*, 2008, pp. 389-394.

[12] M. Merdan, T. Moser, D. Wahyudin, and S. Biffl, "Simulation of Workflow Scheduling Strategies Using the MAST TMS," *Intl Conf on Control, Automation, Robotics and Vision*, 2008, pp. 1172-1177.

[13] M. Pechoucek, and V. Marík, "Industrial deployment of multi-agent technologies: review and selected case studies," *Intl J on Autonomous Agents and Multi-Agent Systems*, vol. 10, no. 3, 2008, pp. 397-431.

[14] N. Ruiz, A. Giret, and V. Botti, "Towards an Agent-based Simulation Tool for Manufacturing Systems," *ETFA '06. IEEE Conference on Emerging Technologies and Factory Automation, 2006.*, pp. 797-804.

[15] W. Shen, L. Wang, and Q. Hao, "Agent-Based Distributed Manufacturing Process Planning and Scheduling: A State-of-the-Art Survey," *IEEE Transactions on Systems, Man and Cybernetics. Part C Applications and Reviews*, vol. 36, no. 4, 2006, pp. 563.

[16] K.P. Sycara, "Multiagent Systems," *AI Magazine*, vol. 19, no. 2, 1998, pp. 79-92.

[17] P. Vrba, V. Marík, and M. Merdan, "Physical Deployment of Agent-based Industrial Control Solutions: MAST Story," *2008 IEEE International Conference on Distributed Human-Machine Systems*, 2008, pp. 133-139.

[18] W.J. Zhang, and S.Q. Xie, "Agent technology for collaborative process planning: a review," *The International Journal of Advanced Manufacturing Technology*, vol. 32, no. 3, 2007, pp. 315-325.