Holonic Manufacturing Execution Systems

P. Valckenaers[1], H. Van Brussel[1] (1)
[1] K.U.Leuven – PMA, Dept. of Mechanical Engineering,
Katholieke Universiteit Leuven, Leuven, Belgium

Abstract
This paper presents the design of a holonic manufacturing execution system. The design is an instantiation of the PROSA reference architecture [1] augmented with coordination and control mechanisms inspired by natural systems – i.e. food foraging behavior in ant colonies. Research prototypes are implemented as multi-agent systems. The main coordination and control mechanisms ensure that the process plans are properly executed and emergently forecast the workload of the manufacturing resources as well as well as lead times and routings of the products. The design empowers the product instances to drive their own production; the coordination is completely decentralized. In contrast to many decentralized designs, the manufacturing execution system predicts future behavior and proactively takes measures to prevent impending problems from happening. A social control mechanism ensures that product instances adhere sufficiently to their declared intentions, which is necessary to guarantee adequate forecast accuracy. The design has been applied to an industrial test case, and the paper discusses results of this case study.

Keywords:
Holonic, Agent, Manufacturing execution system

1 INTRODUCTION

A manufacturing execution system (MES) handles factory operations. It supervises the process control systems, it decides about the routes that products follow through the production system, and it decides when and where operations on products start. Moreover, It must handle all possible processing outcomes, some of which disturb the manufacturing operations significantly – e.g. failures. The wording manufacturing control commonly denotes the task performed by a manufacturing execution system.

Manufacturing control is a daunting task because of the non-linear nature of the underlying production system, the uncertainties stemming from the environment and the production processes, and the combinatorial growth of the decision space. In spite of immense efforts, hierarchical planning-based manufacturing execution systems have been unable to answer the challenges satisfactorily until today. Schedules and plans, originating from higher levels in a manufacturing organization, are known to become ineffectual within minutes on a factory floor.

Investigating an innovative approach to addressing these concerns, researchers have proposed and developed heterarchical manufacturing systems [2]. In heterarchical systems, intelligent products and parts drive their own production in cooperation with intelligent manufacturing resources. However, current heterarchical manufacturing systems yearn for the ability to plan ahead in time without suffering from the disadvantages of conventional control.

Research into holonic manufacturing systems addresses the above issues as well [3][4][5]. With heterarchical designs, holonic manufacturing systems share the highly decentralized product-driven manufacturing control in combination with intelligent equipment. In contrast, holonic systems support flexible hierarchies, which can be formed dynamically through aggregation. Moreover, some holonic systems comprise planners as system components (i.e. hybrid designs). However, the distinction between holonic and heterarchical systems is mostly irrelevant for the research results that are discussed in this paper; the aspects addressed comply with a heterarchical design.

This paper first discusses research by others investigating heterarchical manufacturing systems. Next, it presents the holonic MES design, which is a predictive heterarchical manufacturing system anticipating the consequences of the currently envisaged decisions by the MES and preserving the heterarchical system advantages. Finally, some results of an industrial case study are presented.

2 RELATED WORK

Early work

Duffie [2] pioneered heterarchical manufacturing control. In those early days of computer networks, product parts routed themselves through a manufacturing system and got themselves processed on workstations without much planning or coordination, somewhat similar to the manner in which car traffic operates.

The key advantage of Duffie's design is the much-reduced exposure of the software components in the system; those self-managing parts and processing stations avoid relying on assumptions outside their own scope. E.g. software handling a product part makes no assumptions on the configuration of the factory or the availability of the manufacturing resources. More conventional approaches generally build on assumptions that fail to uphold under real-life factory conditions. Any enhancement on the early designs ought to preserve this crucial property of limited exposure for the system building blocks.

Market/negotiation-based manufacturing control

The early work was criticized in the past for being unable to offer performance guarantees (i.e. resembling car traffic); more recent work from Duffie is addressing this concern [6]. Additionally, the predictability of more conventional designs is now generally accepted to be an illusion as well (weakening the criticism). To remedy this lack of focusing on performance, researchers have developed more elaborate designs. These are generally based on market mechanisms and negotiation protocols.

Lin and Solberg added a market mechanism to allocate the available production capacity more optimally [7]. AARIA is a comprehensive framework to use negotiation protocols and pricing for allocating production capacity along a production chain [8]. Holonic manufacturing systems have embraced market mechanisms as well [9]. Bussmann also uses negotiation protocols to allocate work pieces to workstations but uses queue length, buffer space and process availability instead of price [10]. The latter system is implemented in an existing factory within the automotive industry. Typically, all these systems use a variation on the well-known contract net protocol.

These enhancements generally preserve the configure-ability of the early designs but the performance-enhancing additions are less robust. Price (or what is used as price) is unable to capture all relevant information, causing the need for constant tuning of the control parameters under changing conditions. More importantly, the negotiation protocols cannot be easily extended to look ahead in time and only serve to allocate the next processing step when the current processing step is about to finish. The use of deadlines (time-outs) makes the protocols heavy and slow. The complication of handling frequent cancellations, needed to plan ahead in time, would be prohibitive.

Discussion

Overall, these designs are limited concerning the types of manufacturing systems that they can manage effectively. Typically, they manage single parts moving through a factory with mutually equivalent processing stations and supporting flexible routing. Basically, these are systems in which myopic decisions can be made near optimal. The remainder of this paper introduces a predictive heterarchical design, preserving the limited exposure property of the early systems.

3 PREDICTIVE HOLONIC MANUFACTURING

This section discusses the predictive MES. It presents the embedded heterarchical subsystem followed by a discussion of the predictive coordination mechanisms.

3.1 Holonic manufacturing control

The manufacturing control system adopts the standard approach for heterarchical designs in which every relevant physical entity in the manufacturing system is reflected in the control system software. For instance, machines and product parts each have a corresponding computing agent in the control system. The heterarchical subsystem implements the PROSA reference architecture [1]. Systems designed along this architecture are composed of three types of basic agents: order agents, product agents, and resource agents. These basic agents are structured using object-oriented concepts like aggregation and specialization. Optional staff agents, like in human organizations, can be added to assist the basic agents with expert knowledge (e.g. schedulers).

This reference architecture generalizes the standard heterarchical design by reflecting more abstract notions in the manufacturing system. In particular, order agents reflect tasks that do not necessarily correspond to a specific product part in the manufacturing system (for instance, order agents handling preventive maintenance). PROSA also separates technological and logistic concerns. Product agents correspond to product types while order agents correspond to product instances. A full discussion of PROSA and its merits is outside the scope of this paper. The reader is referred to [1][11][12] for more detailed information and in-depth discussions.

Stigmergy infrastructure

The basic PROSA design has been augmented to support stigmergy. Grassé introduces this word to describe how signs in the environment are used to coordinate activities of social insects, replacing direct communication [13]. The display of goods with price labels in shops is an example; road signs are another example of stigmergy.

To support stigmergy, information spaces are attached to the basic PROSA agents. In the present context, only information spaces attached to resource agents are relevant. Any agent acquainted with a resource agent is able to place, retrieve and modify software objects on its attached information space, somewhat analogous to medical staff using clipboards attached to hospital beds. The resource agent has no explicit control or responsibility on what other agents do with its information space. Any software object and its associated retrieval keys can be placed on the board. Note that this protects the resource agent from any software maintenance requirements originating from its visitors (limited exposure).

The information on these boards has a finite lifetime. When the information progresses beyond a given age, it is discarded. Agents must refresh information on these boards fast enough if they want it to remain available for other agents observing the information space. This is a generic mechanism to handle changes: any stale information simply disappears when it becomes too old. The frequency at which information is refreshed and the upper bound on its lifetime determine how fast the system will observe changes. Hence, a system designer trades communication and computing effort against the delay at which changes become known.

Ant agents

Historically, the holonic MES design originated out of inspiration by nature. More specifically, the coordination through stigmergy in food-foraging ant colonies triggered the sequence of design steps leading to the current design. However, this design has evolved significantly and, presently, the analogy with the social insects confuses rather than helps the discussion. Nonetheless, the terminology still reflects this source of inspiration. In an ant colony, ants deposit information in their environment (pheromones) informing other ants about remote facts (how to find food). The PROSA agents create agents for similar purposes (i.e. system-wide coordination; see below). These agents are called ant agents or simply ants in the remainder of this text.

In the current design, ant agents are linked to a resource agent, which is their current position. The ant agents query their resource agent about its connections to neighboring resource agents and use this to virtually navigate through the manufacturing system, more precisely through the network of resource agents reflecting the manufacturing system. Ant agents get their initial position at creation.

Ant agents typically originate from basic PROSA agents, and perform an information retrieval and dissemination task on behalf of the PROSA agent. Ant agents are created at a given rate, among others, to refresh information, before it expires, on the information spaces. Section 3.2 discusses two types of ant agents rendering the MES predictive. First, feasibility ants, another type of ant agent, are shortly discussed below.

Feasibility ants constitute a first type of ant agents in the heterarchical MES. Feasibility ants put signposts on the information spaces enabling order agents to decide locally which routing options are available to them. The ants start at factory entries and/or exits and traverse the network of resource agents while collecting information on the processing capabilities of the resources. At routing opportunities, this information is merged with the available information on the local information spaces. This activity is performed at a regular frequency such that changes become visible throughout the system with a small delay (e.g. 10 sec). A more detailed discussion is given in[14].

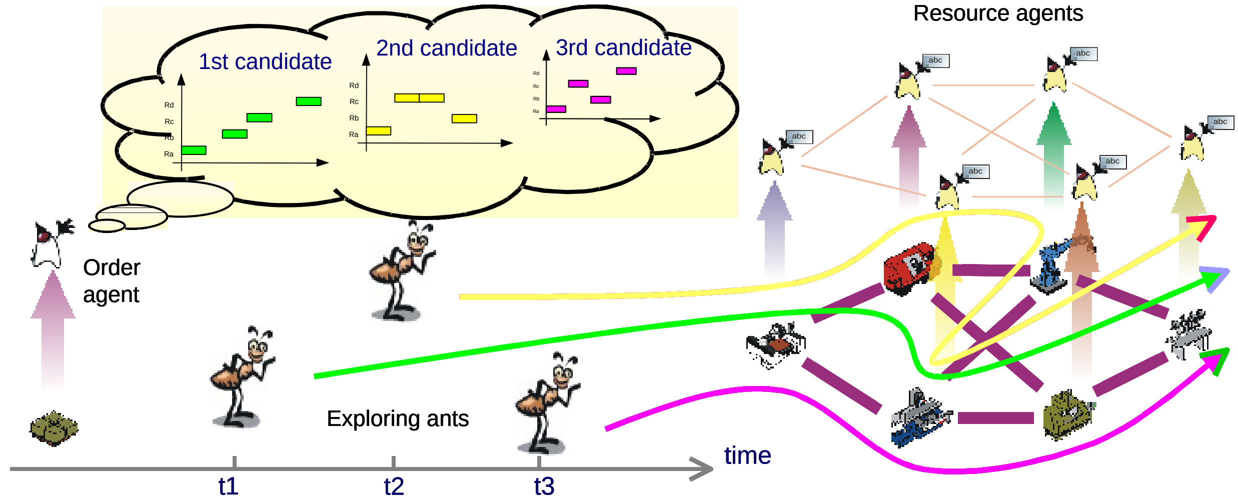

Figure 1: Ant agents scout for solutions on behalf of their order agent.

3.2 Predictive holonic manufacturing control

This section discusses how a heterarchical manufacturing control system is able to account for the near future. Order agents, resource agents and their ant agents perform a choreography from which short-term forecasts materialize.

Exploring ants

Order agents create at a regular frequency ant agents that scout for possible solutions. An exploring ant generates one feasible solution while traveling virtually through the factory and making the resource agents virtually perform the required processing steps. The ant is created at the location of its order agent. It queries the associated resource agent about the ongoing activity. For instance, the order is queued on a conveyor belt and the conveyor belt agent reports the estimated time when the order will reach the end of the belt. The ant agent virtually moves to the end of the conveyor belt and progresses its virtual clock to this estimated time. At the end of the conveyor belt, the exploring ant retrieves the signposts, placed by the feasibility ants, and presents them to the associated product agent to learn the available routing options. The exploring ant selects one of the available options and continues its virtual journey (cf. Figure 1). When the exploring ant arrives on a processing unit, it retrieves the processing capabilities from the resource agent and presents them to the product agent to discover which processing steps can be performed. In addition, the signpost information is used to decide whether leaving the processing unit is already an option. Again, the ant agent selects and virtually executes an option that is available.

The option selection mechanism of the exploring ants is a plug-in for the MES; it is a software component that is considered external to the core of the MES (i.e. not part of the research contribution). This part of the control system is likely to be system-specific and subject to regular maintenance. Note also that well-designed order agents doubtlessly generate exploring ants with a variety of selection mechanisms (adventurous, assertive, following a plan…). Also, exploring agents may use the stigmergy infrastructure to coordinate their exploration among each other and with the resource agents.

Importantly, exploring ants rely on resource agents to provide sound estimates for the duration of transport and processing steps. To this end, resource agents have a 'reservations department' that answers queries about capacity availability. Note that the resource agents mainly require self-knowledge to provide such reservation service to their prospective visitors. Any decision making in the self-scheduling of the resource must be a plug-in. However, this reservations department cannot answer queries properly without an informed estimate of future usage; it needs future visitors to book (make a reservation). The intention ants perform this task.

Intention ants

When an exploring ant finds a solution, it reports back to the order agent. This order agent evaluates the performance of this solution (e.g. rush orders rank solutions by their estimated finishing time). Again, this evaluation is a plug-in, and order agents in a single system doubtlessly will use a variety of mechanisms (for rush orders, normal orders, maintenance orders). Each order agent maintains a small collection of attractive solutions. This collection is selected based on performance and complementarities of the solutions. Exploring ants refresh these candidate solutions regularly.

At a given moment, again chosen by a plug-in, the order agent selects the most attractive solution to become its intention, and starts to create intention ants at a regular frequency (cf. Figure 2). Intention ants behave in the same manner as the exploring ants except for two aspects. First, the option selection mechanism is to follow the order intention. When the intention ant reports back, the order agent observes the consequences of any changes in the system (e.g. a machine breakdown). Second, the ant informs the resource agents on its journey of the order intentions. In other words, it books the required capacity on the resource. In turn, the resource agents receive the necessary information to calculate a short-term forecast of their utilization. This enables the resource agents to give accurate answers to the queries by exploring and intention ants alike. Importantly, bookings must be refreshed regularly. Otherwise, the reservation is discarded and becomes available for other users.

This combination of exploring and intention ants provides both order agents and resource agents with short-term forecasts, enabling predictive heterarchical control. The choreography ensures that agents have limited exposure (make minimal assumptions) safeguarding the advantages of heterarchical designs. Getting these predictions without the exposure typical in planning-based systems is the core contribution of the research presented in this paper.

Socially acceptable behavior

The above interactions are only able to generate a reliable forecast if the order agents adhere sufficiently to declared intentions agents. To ensure this, the MES imposes

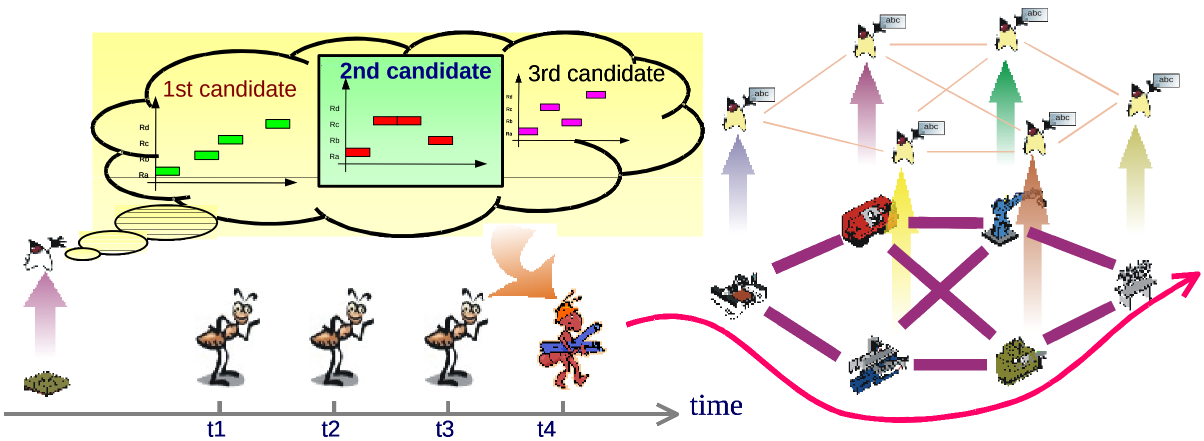

Figure 2: Order agent has selected the second solution and its i ntention ant reserves time slots on resources on its behalf.

socially acceptable behavior upon the members of its agent society. More specifically, order agents can only change intentions in a controlled manner.

First, agents only change their intention if the perceived amelioration is significant. Moreover, this threshold for the perceived improvement is higher for intentions related to a more immediate future. This is similar to the way humans uphold their promises concerning meetings, work, etc.

Second, the order agents change their intentions probabilistically. For instance, when a machine breaks down, all the affected order agents want to switch to alternative machines. However, these agents will only change with some probability during the next refresh cycle. As a consequence, only a fraction changes to the alternative. As more refresh cycles pass, the build-up of a queue in front of the alternative machines becomes visible and after some time, order agents detect that waiting for the repair has become their best option. This mechanism prevents that disruptions create stampedes.

Cooperation with planning systems

When the system enjoys the presence of a good planning system [15], the order agents attempt to execute the plans in three ways. First, a significant fraction of the exploring agents will simply explore the scenario of the plan. Second, adherence to the plan is part of the performance evaluation criteria such that orders will only deviate from the plan if there is a significant perceived benefit. Third, the order agent considers the plan to be its original intention, which triggers the resistance to deviate from the plan by the socially acceptable behavior described above.

Conversely, the holonic MES is able to feed the planning system with a short-term forecast, estimating the state of the manufacturing system when its next plan will become available. A reactive scheduling system can use this estimated future state to initialize its data model for the subsequent run of its planning algorithm.

3.3 Computational complexity issues

The decision-making mechanisms in an MES perform scheduling and schedule execution, inherently solving an NP-hard problem. They face a combinatorial explosion or worse. In practice, this means that any known computationally efficient mechanism only solves the given problem approximately, simplifying the original problem.

In conventional approaches, scheduling systems require human experts to fit the given problem into the data model of the scheduling mechanism (e.g. LP or Lagrange-relaxation). This transformation simplifies and modifies the real-world scheduling problem into an approximation that can be solved efficiently. Note the human experts ensure that the critical aspects are captured while performing the required modeling. Unfortunately, the critical aspects of a manufacturing system are the most likely to change (e.g. future investments increase capacity at the bottleneck). Thus, these human experts need to maintain the data model, which is corroborated by the fact that industrial application of these technologies is almost exclusively found in large capital-intensive industries where superior scheduling easily pays for recurring maintenance efforts. In contrast, the decentralized approaches use a brute-force solution for the combinatorial explosion (cf. section 2) in that they only decide about the next processing step, limiting the range of production systems to which they can be applied (often requiring expensive equipment).

The specific research contribution of the holonic MES in this paper is the innovative manner in which it escapes the combinatorial explosion. The system does not schedule but delegates this task to the plug-ins, which are allowed only short and bounded computational efforts. If a plug-in wants to use complex algorithms, these algorithms must be implemented in a separate planning system, and the plug-in must cooperate with it as described above.

In contrast, the holonic MES focuses on reducing and virtually eliminating the recurring data model maintenance effort, which is the proverbial Achilles' heel of more conventional scheduling in industry. To achieve this, the MES supports a model that reflects the underlying production system closely without approximations to fit a specific scheduling mechanism. Importantly, this model is built from components that can be reused wherever the corresponding entity in reality exists without maintenance. Furthermore, the latter property is maintained when the short-term forecasting mechanism is added.

Such maintenance-free compose-ability requires more computational effort than a centralized monolithic model would. Nonetheless, the holonic design remains efficient. The number of physical entities – machines, product parts – constitutes an upper bound on the number of basic PROSA agents. Each of the agents consumes a bounded amount of processing capacity and communication bandwidth (since they do not address any NP problems). The main activity of order agents is to create and manage exploring and intention ants. The main activity of resource agents is to handle queries from the visiting ant agents.

The number of physical agents multiplied by the refresh rate limits the number of ant agents. The refresh rate for feasibility determines how fast changes are detected (e.g. 1 Hz). The refresh rate for exploring and intention ants determines the optimization effort. In practice, users will select the highest rate possible. The design delivers best results for the selected effort. The design ensures that a

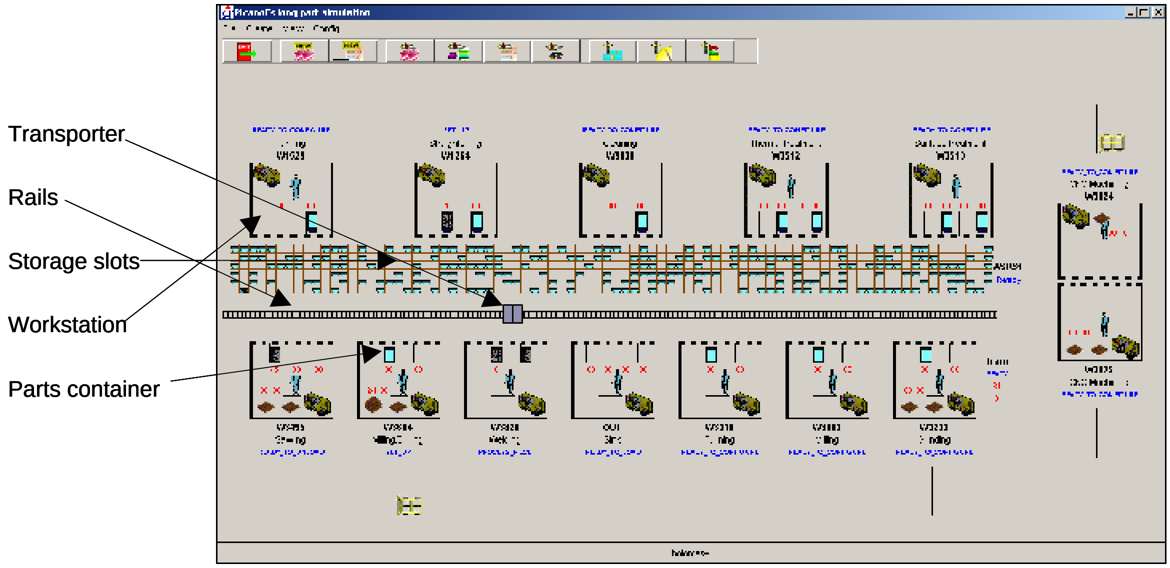

Figure 3: Screenshot from the MPA test case application.

solution is known virtually at any time while, given more time, better solutions are sought for continuously. Overall, the design has low-polynomial complexity.

4 INDUSTRIAL APPLICATION

The predictive heterarchical manufacturing control has been developed in cooperation with industry providing requirements and specifications. Early developments addressed a car body paint shop. Those designs handle homogeneous shop layouts well but fail to cope with the complex heterogeneous nature of the actual paint shop. The lacking feature was the 'socially acceptable behavior' discussed earlier. Subsequently, developments addressed a job shop producing weaving machine parts (Figure 3) in which the 'socially acceptable behavior' was introduced. Initial results reveal how the forecasting operates as expected [12]. Currently, research addresses a network of heat treatment factories, which requires the control to optimize batch building across factory boundaries.

The goal of the research is to provide the industrial user with the means to incrementally enhance his decision-making mechanisms where the core result of the research provides the information, including up-to-date short-term forecasts, on which to construct these mechanisms. This section illustrates how this works out for the case study on the production of weaving machine parts. The crucial aspect of this industrial job shop is to optimize the use of a transport system, called the tram. This tram carries containers with product parts between workstations and storage slots. During periods of heavy demand for transportation (rush hour), it is a bottleneck and causes workstations and operators to idle, which is expensive.

	Tram	W3824	W3310	Total
avg. wait time	76	41	145	287
total wait time	4315	370	435	8928
max wait time	891	84	365	1587
utilization rate	8%	91%	89%	

Table 1: Results from the basic controller.

In a first experiment, the basic design described earlier is applied to control the job shop. The construction of this basic controller occurs in a straightforward plug-and-play fashion. Existing modules and decision packages are used, without any modifications to fit the specific case. With a minimal amount of programming work, putting these modules together, a working online MES is created. All orders are capable to find a solution. Results on the performance of the transport system and the bottlenecks in the plant are reported in table 1. W3824 and W310 are the bottleneck workstations in the system. The rightmost column displays aggregated results for all workstations.

	Tram	W3824	W3310	Total
avg. wait time	109	31	129	274
total wait time	4374	280	387	8488
max wait time	838	57	345	1587
utilization rate	6%	93%	90%	

Table 2: Results from the enhanced controller.

In a second experiment, the product agent is modified such that it becomes knowledgeable of the fact that visiting the storage space is optional. No other part of the control system requires adaptation. Note that in the current industrial system (in reality), this enhancement cannot be implemented because the effects of direct workstation-to-workstation transports on the overall system performance are unclear (due to lacking up-to-date forecasts) and therefore represent an unacceptable risk. Table 2 shows the improvement obtained by these changes.

The main effect on the tram is a reduction of its load by 25% and the number of transports by 30% resulting in reduced waiting times for all workstation and especially the bottleneck stations. The waiting times on the latter stations are reduced by 24% and 11% respectively. In turn, the utilization rate on these bottleneck stations increases, implying that system throughput increases as well. This improvement is smaller (2% and 1%) because machining operations last longer than transportation.

The experiments on this test case are ongoing research. The next improvement on the agenda is to reduce the number of shifts on the non-bottleneck stations. Based on the forecast information, the system can identify opportunities to move work out of one shift into another without disturbing the remainder of the system, especially at the bottlenecks. The aim is to cluster operations within shifts, rendering other shifts completely empty to reduce personnel costs.

Subsequently, the excess capacity on the tram, outside rush hours, will be used to prepare the work during periods of high demand. The availability of an up-to-date prediction is essential for this enhancement since it both informs the system whether there is an opportunity to rearrange the storage (a period of low demand or no demand lies ahead) and tells the system which rearrangement is likely to lower the workload during upcoming periods of high demand.

Note also that this approach offers a medium to coordinate the main operations with secondary operations. For instance, the short-term forecasts can be used to plan and control the transport of tools and fixtures to the machining stations in the test case, which utilizes the tram as well. The holonic MES allows operators to select times at which they do not disturb production, if possible. Conversely, the operator can inform the MES of his intention to use the tram for tool/fixture transport such that the remainder of the system is able to account for this.

Overall, the experimental results corroborate that the holonic MES design complies with the industrial requirement to be able to enhance gradually the manufacturing control system, and especially the decision-making mechanisms therein, while addressing the concern of their choosing in the manufacturing operations.

5 CONCLUDING REMARKS

This paper presents a holonic MES that preserves the advantages of heterarchical designs and predicts the near future while accounting for changes and disturbances. To achieve this, the software agents reflect the underlying manufacturing system (e.g. order agents reflect tasks) and delegate consistently. For instance, exploring ant agents query the associated resource agent about processing times; they do not have their own model of the resource and therefore make no assumptions that can be faulty. The forget-and-refresh mechanism of the stigmergy infrastructure ensures that information remains up-to-date.

This reflection of the manufacturing reality is extended to the intentions of the order agents. This yields the short-term forecasts. Imposing a social control mechanism ensures the quality of these forecasts. The actual decision-making software components cannot be implemented without losing the advantages of the original heterarchical designs, and remain application-specific plug-in components.

The contribution of the research discussed in this paper is foremost an innovative system architecture, adding the ability to look into the future to a decentralized MES. Consider the following analogy in the navigation domain. Here, planning systems generate optimized route descriptions and industrial application of the technology suffers from the explosion of the number of required route descriptions as well as the fragility of these descriptions. A navigation map escapes the above problems by reflecting the world and by leaving the decision-making issues to ‘plug-ins’. Indeed, users still have to generate the routes that they require. Notice that if researchers in the navigation field were using a common benchmark (e.g. the Paris to Rome route), a map will not deliver any scores for such a benchmark. It intrinsically is a halfway solution.

For MES, the research results in this paper delivers such maps, which not only represent the current situation in the production system but include a short-term forecast of future states as well. These maps provide a coordination medium, which multiple planners can use to cooperate (e.g. operators can plan ahead when to perform cleaning tasks). There is no need to reconcile data models when multiple planning systems are used. Indeed, like in navigation maps, information can be missing but there can never be a conflict because the world being reflected is coherent and consistent. It suffices to look closer at the world to find out which error causes a conflict; there is no need to make a choice.

6 ACKNOWLEDGMENTS

This paper presents work funded by the Research Fund of the K.U.Leuven (Concerted Research Action on Agents for Coordination and Control) and the European Commission (EU projects Mascada, MPA and MABE).

7 REFERENCES

[1] Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P., 1998, Reference architecture for holonic manufacturing systems: PROSA. Computers In Industry, 37:255-274.

[2] Duffie, N., 1990, Synthesis of heterarchical manufacturing systems. Computers in Industry, 14:167-174.

[3] Koestler, A., 1967, The ghost in the machine. Hutchinson & Co, London.

[4] Gou, L., Hasegawa, T., Luh, P.B., Tamura, S., Oblak, J.M., 1994, Holonic planning and scheduling for a robotic assembly testbed, Proc. of the 4th Int. Conf. on CIM and Automation Technology, IEEE, 142-149.

[5] Valckenaers, P., Bonneville, F., Van Brussel, H., Bongaerts, L., Wyns, J., 1994, Results of the holonic manufacturing control system benchmark at K.U.Leuven, Proc. of the 4th Int. Conf. on CIM and Automation Technology, IEEE, 128-133.

[6] Prabhu, V., Duffie, N., 1995, Modelling and analysis of non-linear dynamics in autonomous heterarchical manufacturing system control, CIRP Annals, 44/1:425-428.

[7] Lin, G. Y., Solberg, J. J., 1992, Integrated shop floor control using autonomous agents, IIE Transactions, 24/3:57-71.

[8] Parunak, H.V.D., Baker, A.D., Clark, S.J., 1997, The AARIA Agent Architecture: An Example of Requirements-Driven Agent-Based System Design. Proc. 1st Int. Conf. Autonomous Agents, 482-483.

[9] Márkus, A., Kis, T., Váncza, J., Monostori, L., 1996, A market approach to holonic manufacturing. CIRP Annals, 45/1:433-436.

[10] Bussmann, S., Schild, K., 2000, Self-Organizing manufacturing control: an industrial application of agent technology. Proc. 4th Int. Conf. on Multi-Agent Systems, 87-94.

[11] Wyns, J., 1999, Reference architecture for Holonic Manufacturing Systems - the key to support evolution and reconfiguration, Doctoral thesis, K.U.Leuven, people.mech.kuleuven.ac.be/~jwyns/phd/order.html

[12] Valckenaers, P., Saint Germain, B., Verstraete, P., Hadeli, Zamfirescu, C., Van Brussel, H., 2004, Ant colony engineering in coordination and control: how to engineer a short-term forecasting system. Proc. IWES’04, 125–132, selected for publishing in CIRP Journal on Manufacturing Systems, Vol. 34, 2005.

[13] Grassé P.P., 1959, La theorie de la stigmergie: Essai d’interpretation des termites constructeurs. Insectes Sociaux, 6:41–83.

[14] Steegmans, E., Holvoet, T, Janssens, N., Michiels, S., Berbers, Y., Verbaeten, P., Valckenaers, P., Van Brussel, H., 2002, Ant algorithms in a graph environment: a meta-scheme for coordination and control, AI and Applications, ACTA Press, 435-440.

[15] Csáji, B., Kádár, B., Monostori, L., 2003, Improving multi-agent based scheduling by neurodynamic programming. Lecture Notes on AI, 2744: 110-123, Springer.