Robust Planning and Control Using Intelligent Products

Gerben G. Meyer and J.C. (Hans) Wortmann

Department of Business & ICT, Faculty of Economics and Business, University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands
{g.g.meyer, j.c.wortmann}@rug.nl

Abstract. The advances in production planning and control in the last decades have confirmed the centralized nature of the planning function. However, centralization has disadvantages when quick response to local production problems is required. Therefore, renewed interest in decentralized approaches emerges. This paper investigates the possibility to use intelligent products for decentralized handling of disturbances. Intelligent products are aware of their local context and they can negotiate with local manufacturing resources. Therefore, local solutions for problems can be suggested, virtually at the same time at which the problem occurs. The paper demonstrates the viability of this approach via a simulation study. For reasons of comparison, the TAC SCM environment is used. Moreover, robustness is taken as an additional measurement of performance. The results of the simulations are encouraging.

1 Introduction

Advances in production and supply chain planning over the past decades have steadily resulted in centralization of the planning function. There are good reasons for this centralization, both from a material perspective and from a capacity perspective. From materials perspective, coordination over the supply chain reduces the bullwhip effect [8,11]. When combined with proper rules for safety stocks and lot sizes, this effect may almost be eliminated. Moreover, the problem of matched sets of parts in assembly requires coordination of supply streams for all components in the bill-of-material [14], which seems again to justify centralized planning. From capacity perspective, optimization of one resource will usually impact other resources, such that some kind of coordination is not only useful but nearly unavoidable.

However, centralized planning also has its drawbacks, as for example is shown by [4]. These drawbacks appear in practice, and are caused by the many small disturbances that occur in manufacturing and transportation. A typical example of such a small disturbance is when a component is damaged, although it was planned to be used in manufacturing. In this case, a similar component needs to be sourced from somewhere else in order to continue with the original plan. Often, these kind of disturbances are not even made known to the central planners, as they are often solved on a more local level by for example a foremen. Other kind of disturbances can include production errors and misshipments. These disturbances are one of the many causes why central plans in factories are seldom realized. Therefore, this paper proposes a more robust planning and control system, based on the concept of intelligent products, which goal is to be able to handle these disturbances in a more effective way.

The performance of the proposed system will be compared with other approaches, using the Trading Agent Competition Supply Chain Management (TAC SCM) simulated supply chain [3]. However, the usual measurement of performance in TAC SCM are the financial results, in terms of costs made and penalties paid balanced against profits made in sales. In contrast, this paper argues that such a measurement of performance does not reflect the impact of disturbances enough. More fundamentally, simulation studies tend to ignore the disturbances, although they dominate the planner's activities in practice. This paper aims to contribute here by proposing robustness of a planning and control system as an additional measurement of performance.

The paper is structured as follows. In the following section, the concept of intelligent products is elaborated. Next, the applied methodology and the proposed planning and control system design are discussed in more detail. Afterwards, the performance results of the proposed system compared to other systems are presented. Discussion and conclusions are provided in the last sections.

2 Background

Nowadays, there is an increasing interest in the field of intelligent products, and how intelligent products can be applied in different fields, such as in manufacturing and supply chain management [13]. McFarlane et al. define an intelligent product as a physical and information-based representation of a product [12]. Figure 1 shows an example of such a product. In this figure, the jar of spaghetti sauce is the physical product, the information-based representation of the product is stored in the database, and the intelligence is provided by the decision making agent. The connection between the physical product and the information-based representation is made using a tag and a reader, as will be further discussed later on. The fundamental idea behind an intelligent product according to Kärkkäinen et al. is the inside-out control of the supply chain deliverables during their life-cycle [10]. In other words, the product individuals in the supply chain themselves are in control of where they are going, and how they should be handled.

Recent technologies, such as automatic identification (Auto-ID), embedded processing, distributed information storage and processing, and agent based systems have been the main enablers for intelligent products. Auto-ID technologies, such as barcode and RFID, are commonly used to uniquely identify individual products or delivery units. Especially RFID tags are suitable for tagging individual products, as multiple RFID tags can easily be read simultaneously, without requiring a line-of-sight, such as is the case with barcodes. In addition to automatic identification, Auto-ID technologies often also include localization and

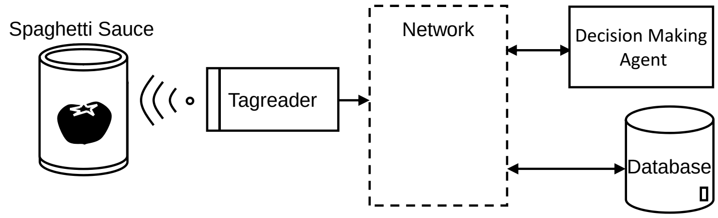

Fig. 1. An intelligent product [21]

sensor technologies. Localization techniques, such as GPS, are often combined with automatic identification, as the location information is useless without the identity of the located entity [19]. Another frequently applied technique is updating the location status of the product at the moment its barcode or RFID-tag is scanned, when the physical location of the scanner is known [7].

The vision of intelligent products is to seamlessly connect the products in the physical world with their representation in information systems, e.g. through a product agent as proposed by [6]. Intelligent products would make it possible to avoid media breaks between the real word and the digital world. Thereby, data about the current and past state of products from the physical world can be retrieved and updated when needed. The basic building block for implementing a distributed information storage and processing system for products is that products are identified by globally unique identifiers that either encode links to information sources directly or that can be used as look-up keys in some kind of network infrastructure. The main three currently known approaches for distributed information storage and processing are EPC Global [17], ID@URI [7], and WWAI (www.wwai.org). A technical analysis and comparison of these approaches can be found in [5].

Agents are a useful paradigm to implement the intelligence part of intelligent products. There are several reasons why the use of an agent-based platform for intelligent products is beneficial. Firstly, when there is a high number of products, the number of products in need of explicit control from the user has to be reduced. This can be achieved by making the products autonomous. In this way, intelligent products with knowledge and reasoning capabilities can do most of the repetitive tasks in an automated way. Secondly, intelligent products should be able to detect and react to changes in the environment. Agents can pro-actively assist the product and try to achieve goals given the change of the environment. Agents can also help in discovering information about the environment by communicating with agents of other products. It is therefore clear that intelligent agents have characteristics which are desirable for intelligent products. Of course, an application for intelligent products can be created without the use of agents, but by using agents, one can take advantage of the methodologies and solutions provided by the multi-agent paradigm [1].

3 Methodology

To compare the performance of the proposed system as will be described in the next section with existing systems, the TAC SCM simulated supply chain is used, as it provides a well-founded testbed for production planning and control systems of manufacturers. In the TAC SCM competition, six manufacturers are competing with each other for customer orders and supplier components. A typical supply chain scenario of a TAC SCM game can be seen in Figure 2. In such a game, manufacturers try to win customer orders for delivering Personal Computers to them. Furthermore, to deliver these PCs to the final customers, manufacturers need to buy components from suppliers, assemble the PCs, and finally ship them. For this purpose, every manufacturer has an identical PC factory containing an assembly cell capable of assembling any type of PC, and a warehouse that stores both components and assembled PCs.

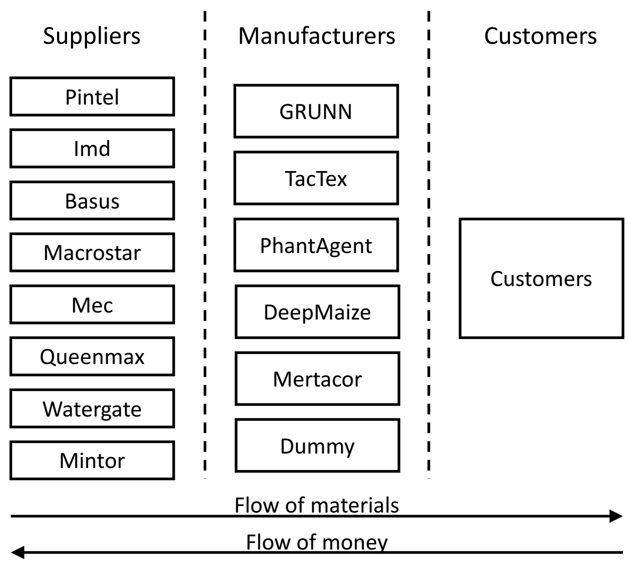

Fig. 2. A TAC SCM scenario

In the current TAC SCM games and competitions, the performance indication of a manufacturer is solely based on the financial result, in terms of costs made for material, storage and penalties paid balanced against profits made in sales. In principle, the manufacturer with the highest bank account at the end of a game wins that game. This measurement of performance gives a good indication of which manufacturer is the most efficient one, in terms of costs and benefits. However, it gives no good indication about the robustness of the manufacturer, in case when the manufacturer has to deal with disturbances. For showing the robustness of a manufacturer, a measurement is needed which only indicates the capability of a manufacturer to handle unexpected disturbances in a flexible way. The financial results of the manufacturers gives an indication of the overall performance, but robustness is only a minor part of that. Therefore, in this paper, an additional measurement is used. This measurement is the percentage of customer orders that are delivered to the final customer in time, i.e. if the delivery is before or on the due date of the specific order. This is considered to be a good measurement for the robustness of a manufacturer, as it gives an indication about the capabilities of a manufacturer to still deliver products to a customer in time, even when disturbances are happening.

Normally, a TAC SCM game does not contain any disturbances by design. There is only some variability between games where manufacturers have to deal with, such as the amount of late supplier deliveries. Therefore, in order to be able to test the robustness of a manufacturer, a typical disturbance which happens frequently in practice has been added to the game. In the slightly altered version of a TAC SCM game, every component which is delivered by a supplier to a manufacturer has n percent chance to be unusable. In such a case, the component will not be added to the inventory of the manufacturer. The reason why the component is unusable reflects to the possibility that in reality a component can be damaged, broken, or misshipped. With this additional disturbance added to the game, experiments have been conducted with three different values for n, namely:

- $n = 0$. In this case, none of the delivered components will be unusable. Therefore, this scenario is the same as the original TAC SCM scenario.
- $n = 5$. In this case, every component has a 5% chance of being unusable.
- $n = 10$. In this case, every component has a 10% chance of being unusable.

For every value of n, a total number of 26 simulations have been conducted, in order to get more significant results. In every simulation, the same opponents have been used, namely: TacTex-07 [15,16], PhantAgent-07 [18], DeepMaize-07 [9], and Mertacor-08 [2,20]. These opponents have been chosen based on their rankings in recent TAC SCM competitions, as well as their availability on the Agent Repository on the TAC website (`www.sics.se/tac`). The sixth manufacturer position was filled by a built-in *dummy* manufacturer. The next section of this paper describes the design of the proposed planning and control system for a manufacturer. Afterwards, the results are presented.

4 Manufacturer Design

This section describes the design of the TAC SCM manufacturer agent, named GRUNN, as it has been used within the conducted simulations. The GRUNN agent can be downloaded from the Agent Repository on the TAC website, as well as from `www.agentlab.nl/tacscm`. In this section, the description of the design is split into two parts, namely the structural design, and the behavioral design.

4.1 Structure

The main structure of the manufacturer agent can be seen in Figure 3. The figure shows a UML class diagram, in which the different internal agents of the manufacturer agent, as well as their relationships. Within the manufacturer agent there are four planner agents, each with different responsibilities, such as purchasing, selling, producing, and shipping. Furthermore, there are component type agents, product type agents, and product agents. Each agent type will be shortly described next.

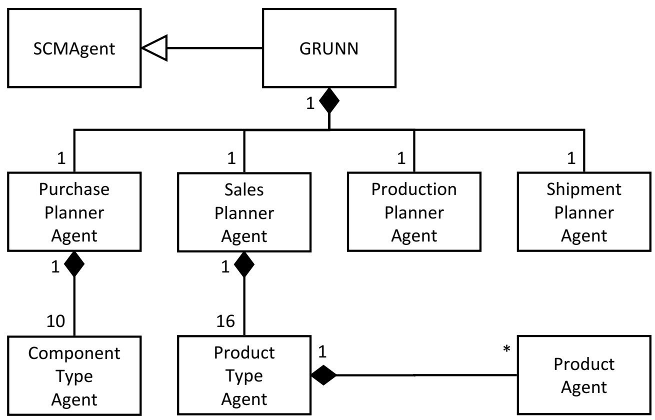

Fig. 3. Class diagram of the internal structure of the manufacturer agent

- The *purchase planner agent* is responsible for acquiring components, which are required for the production of the to be delivered products. However, most of the tasks of this agent are transferred to other agents, as the purchase planner agent creates a separate agent for each component type. This separate agent is responsible for all the tasks related to one particular component type.
- The *sales planner agent* is responsible for acquiring orders. However, most of the tasks of this agent are transferred to other agents, as the sales planner agent creates a separate agent for each product type. This separate agent is responsible for all the tasks related to one particular product type.
- The *production planner agent* is responsible for assigning production capacity to products which are in need of assembly.
- The *shipment planner agent* is responsible for shipping assembled products to the waiting customers.
- A *component type agent* is responsible for acquiring components of one certain type. For this, every component type agent needs to negotiate with the suppliers of this component type.

– A *product type agent* is responsible for acquiring orders of one certain product type. For this, every product type agent needs to negotiate with potential customers.
– A *product agent* is responsible for the complete processing of one final product. In the case of TAC SCM, every customer order is considered to be a product, as every customer order can be seen as an individual and unique product which needs to be delivered by the manufacturer to the customer. Therefore, every customer order will have one product agent assigned to it, which makes the customer order an intelligent product. The responsibility of the product agent includes the procurement of components required for the assembly, the procurement of the required production capacity, as well as arranging the shipment of the finished products to the customer.

4.2 Behavior

This subsection will describe the behaviors of the three most important agent types within the design of the manufacturer agent: the component type agent, the product type agent, and the product agent.

Component type agent Every component type agent needs to acquire sufficient components of one certain type. For this, the behavior of Figure 4 is applied by every component type agent. The figure shows a UML communication diagram, in which the communication of a component type agent with a supplier can be seen. This act of communication consists of three steps, which will be discussed next.

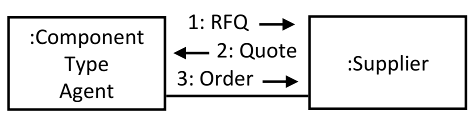

Fig. 4. Behavior of a component type agent

First, the component type agent will send Request For Quotes (RFQs) to every supplier, which can deliver the component type this agent is responsible for. The amount of components as well as the delivery date asked for in an RFQ are based on sales estimations, the quantity that is still in inventory, and the quantity that is ordered but still needs to be delivered. This sales estimation is based on (historical) information which the component type agent receives from the different product type agents. Secondly, suppliers will send quotes back to the component type agent, telling the agent how much they can deliver, on what date, and for what price. Finally, the component type agent will compare the different quotes, and respond by sending orders back to the suppliers who had the best quotes for this component type. Which quote is considered to be the best quote is primarily based on the price per component, but when prices are almost the same it is also based on the quantity and the delivery date.

Product type agent Every product type agent needs to acquire orders for products of one certain type. For this, the behavior of Figure 5 is applied by every product type agent. The figure shows a UML communication diagram, in which the communication of a product type agent with a customer and a product agent can be seen. This act of communication consists of four steps, which will be discussed next.

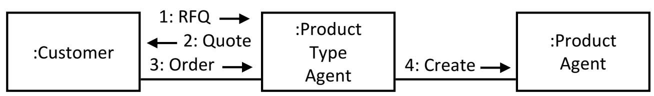

Fig. 5. Behavior of a product type agent

First, the product type agent will receive RFQs of customers, in case customers are requesting quotes for products of the type this agent is responsible for. Each RFQ will contain information about the amount of products, as well as a due date. Secondly, the product type agent will respond with a quote, when the agent considers it feasible to deliver the product before the due date of the customer with a positive financial result. To achieve this, the agent will calculate a price per product based on an estimation of the current market price and adjusted according to the current factory load. This price is compared with the costs of the required components, resulting in a decision whether the quote will be send to the customer or not. Thirdly, when a customer considers the quote of the product type agent the best compared to the other manufacturers, the customer will send back an order. Finally, for every customer order the product type agent receives, a product agent is created, which will be responsible for the complete processing of this one order.

Product agent As mentioned before, a product agent is responsible for the complete handling and processing of one particular order. For this, the behavior of Figure 6 is applied by every product agent. The figure shows a UML communication diagram, in which the communication of a product agent with a component type agent, a production planner agent, and a shipment planner agent can be seen. These communication acts are part of different planning tasks in which the product agent is playing a role. These different planning tasks in which the product agent is involved will be discussed in more detail next.

- The *component planning* is the first planning task in which the product agent is involved. Product agents should be able to assist the component type agent in distributing available components among the different products who require components for production. This functionality requires the

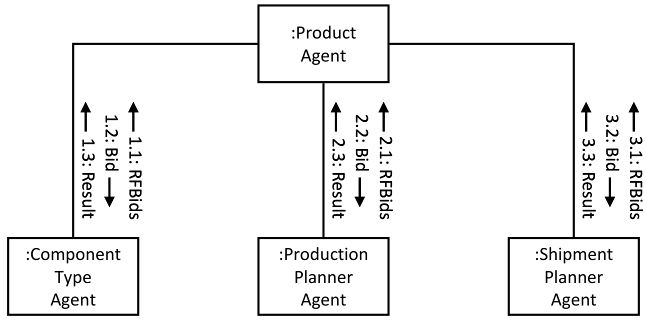

Fig. 6. Behavior of a product agent

intelligent product to already exist before the actual product is produced, i.e. the intelligent product is already in existence from the moment that there is a plan to make the product. This distribution of components among products should be based on priority, therefore, products with earlier due dates should get priority above products with later due dates. In order to achieve a distribution of components based on priorities, an auction based negotiation system is used, which consists of several steps. First, every component type agent will send a Request For Bids to all product agents, when it has components to distribute. Secondly, every product agent who is in need of this component type will send a bid to this component type agent, containing the amount of components of this type it needs, as well as the offered price per component. In this approach, the price per component the product agent is offering will increase when the amount of days left till the due date of the specific order is decreasing. Finally, the component type agent will inform all agents who have send a bid whether they have won the components or not. The product agents with the highest bids will always win the auction, as long as the component type agent has enough components in stock.

– The *production planning* is the second planning task in which the product agent is involved. Product agents should be able to assist the production planner agent in distributing the available production capacity among the different products who require production. As with the component planning, the distribution of production capacity among products should be based on priority, therefore, products with earlier due dates should get priority above products with later due dates. In order to achieve a distribution of production capacity based on priorities, an auction based negotiation system is used, which consists of several steps. First, the production planner agent will send a Request For Bids to all product agents, when it has production capacity to distribute. Secondly, every product agent who is in need of production will send a bid to the production planner agent, containing the amount of production capacity it needs, as well as the offered price per production unit. In this approach, the price per production unit the product agent is offering will increase when the amount of days left till the due date of the specific

order is decreasing. Finally, the product planner agent will inform all agents who have send a bid whether they have won the production capacity or not. The product agents with the highest bids will always win the auction, as long as the production planner agent has enough production capacity available.
- The *shipment planning* is the third planning task in which the product agent is involved. Product agents should be able to assist the shipment planner agent in planning the shipments of finished products to the customers. Differently than the component planning and production planning, no prioritizing is needed, as there is no limitation on the shipment capacity in case of TAC SCM. However, for design consistency, the applied approach assumes a limited shipment capacity, which therefore requires prioritization. In order to achieve a distribution of shipment capacity based on priorities, an auction based negotiation system is used, which consists of several steps. First, the shipment planner agent will send a Request For Bids to all product agents. Secondly, every product agent who is in need of shipment will send a bid to the shipment planner agent, containing the amount of shipment capacity it needs, as well as the offered price per shipment unit. Finally, the shipment planner agent will inform all agents who have send a bid whether they have won the shipment capacity or not. But as there is no limitation on the shipment capacity available, always all product agents with bids will win the auction.

The system as described in this section will not be able to give the best possible plan, as a centralized system can always calculate a closer to the optimal solution within a mathematical domain. However, this system presented here can result in a very robust manufacturer, as will be demonstrated by the results in the next section.

5 Simulation Results

This section shows the simulation results of the conducted experiments. As described in the methodology section, three different experimental setups have been used, namely with zero, five, and ten percent of delivered components which were unusable, and therefore not delivered to the inventory of the manufacturer. For every experimental setup, a total number of 26 simulation runs have been executed, and the results presented in this section are based on the averages of these simulation runs. For the GRUNN agent, the standard deviations are also shown in every graph by means of error bars. The dummy agent is ommited in the results presented in this section, as this agent did not provide any relevant results. However, all detailed results including standard deviations for all agents can be found in the appendix of this paper.

Figure 7 shows the results of the conducted experiments in terms of robustness, i.e. in terms of orders finished in time. The graph shows that the percentage of orders finished in time is decreasing for all agents when the percentage of unusable components is increasing. Only GRUNN is an exception to this. Even in the

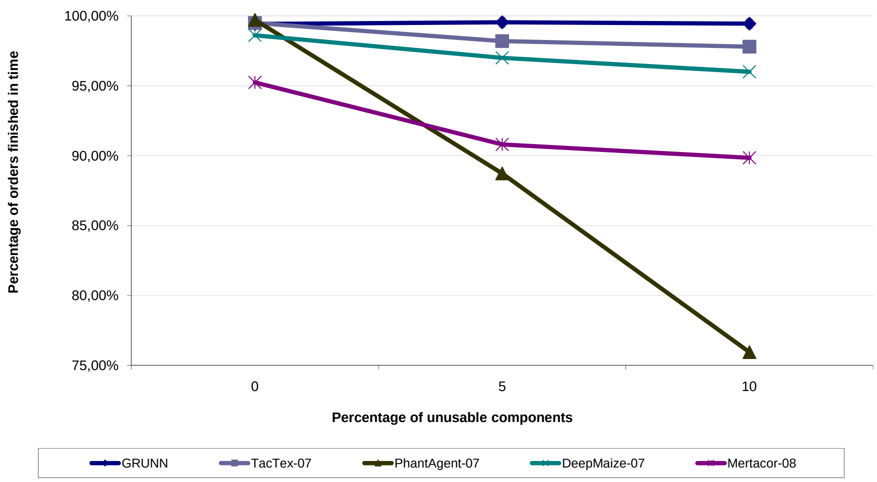

Fig. 7. Performance of manufacturers in terms of orders finished in time

case where ten percent of all components are unusable, GRUNN still manages to finish nearly all orders in time. This observation confirms that an approach based on intelligent products can be very effective in handling disturbances.

Figure 8 shows the results of the conducted experiments in terms of profit. Two important observations can be made from the graph. Firstly, the graph clearly shows that for all three different experimental setups GRUNN does not perform as well as the other agents in terms of profit. This observation is in line with our expectations. Secondly, for all manufacturers, the profit is decreasing when the amount of unusable components is increasing. This observation is also in line with our expectations, as manufacturers need to buy more components to finish the same amount of orders, when the amount of unusable components is increasing.

A very simply way to deal with unusable components in the case of $n = 5$ or $n = 10$ would be to increase the component inventory "safety stock" by a small margin. Figure 9 shows the average storage costs per accepted order for every manufacturer. This gives an indication of the inventory levels of each manufacturer. A simply way to deal with unusable components in the case of $n = 5$ or $n = 10$ would be to increase the component inventory "safety stock" by a small margin. The figure however shows that GRUNN does not hold a significantly larger inventory than the other manufacturers, and therefore does not deal with the unusable components by increasing the safety stock..

6 Discussion

The use of centralized planning systems seems to be justifiable because of many reasons, as was already mentioned in the introduction of this paper. Also the

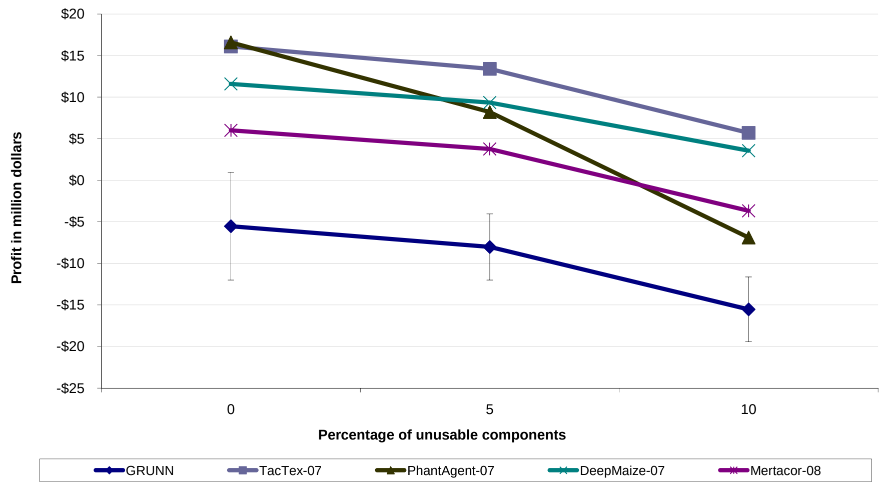

Fig. 8. Performance of manufacturers in terms of profit

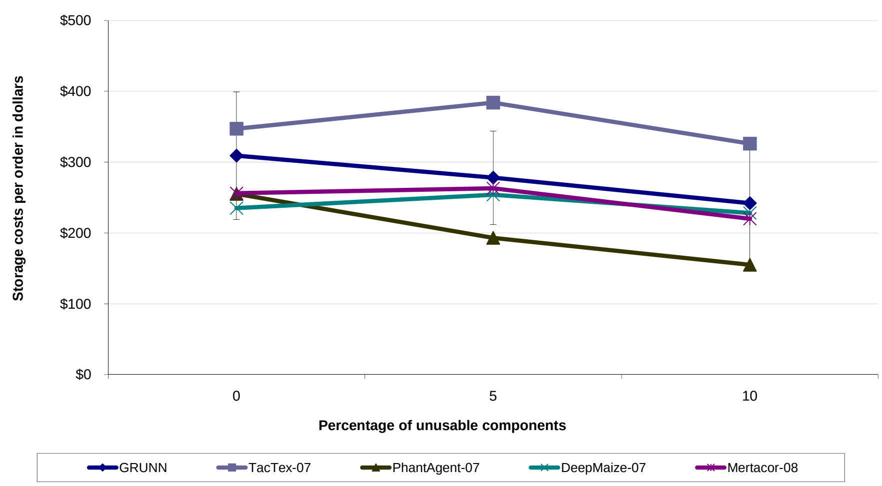

Fig. 9. Storage costs of manufacturers per accepted order

simulation results given in this paper justify centralized planning, as all manufacturers have a higher average profit in TAC SCM simulations than the proposed intelligent product based manufacturer. This is however not a surprise, as in principle a central algorithm can always calculate the optimal solution in a closed and modeled world. On the other hand, the performance of decentralized planning systems are normally greedy, and in that sense suboptimal.

However, during the execution of the plan, unforeseen events can happen. Such disturbances can be solved both in a centralized or distributed way. The results of this paper show that the intelligent products approach is very robust in terms of handling these disturbances. Therefore, this approach seems to be very promising as a control system during the execution of a plan. However, if a simulated environment contains disturbances in a way as they happen in real life, the simulation still only contains modeled versions of these disturbances. In such a modeled environment, a centralized approach can in the end still always outperform a distributed approach, as it is always possible to calculate an optimal solution within a model. Therefore, we argue that solving disturbances locally instead of centrally is especially beneficial outside a simulated environment, because of several reasons. Firstly, a central system will not have enough local attention for individual small problems, and therefore notice and solve problems of individual products too late. On the other hand, intelligent products can notice and solve their own problems locally. Secondly, for a central system it is more difficult to take all local constrains of individual products into account properly. This can be done more easily in a localized way. Finally, a central system always requires a central point of communication. When this central point of communication is down for whatever reason, the whole system fails. A distributed system is more robust against this type of problems. Therefore, we believe that an intelligent products approach will have a bigger benefit in real life situations, where always situations can occur which are outside the model. This could prevent a central system to calculate a good solution when disturbances are happening.

However, as a central system is better in terms of creating an optimal plan, and an intelligent products based approach seems to be better in terms of robustness during plan execution, we believe the "ideal" planning and control system would be a combination of the best of both worlds. Therefore, future work should be focused on investigating how a production planning and control system for a manufacturer can be improved by combining both a centralized approach and a distributed intelligent products approach. This can be studied using the TAC SCM testbed, but it would especially be important to study this in a real world setting.

7 Conclusions

In this paper, a new production planning and control system was presented. This system is based on the concept of intelligent products. A product becomes an intelligent product, when an intelligent agent is attached to the product, which is managing that individual product locally. In such a way, the planning and control system can be more robust, as disturbances can be solved locally. This approach was validated by comparing the performance with other manufacturer systems in the TAC SCM testbed. Conducted simulations in the TAC SCM testbed showed good results in terms of robustness for the planning and control system based on intelligent products, but poor results in terms of profit. Therefore, it requires further investigation how a system can use the best of both centralized planning and distributed disturbance handling.

References

1. C. Carabelea, O. Boissier, and F. Ramparany. Benefits and requirements of using multi-agent systems on smart devices. In *Euro-Par 2003 Parallel Processing*, volume 2790 of *Lecture Notes in Computer Science*, pages 1091–1098. Springer, 2004.
2. K.C. Chatzidimitriou, A.L Symeonidisa, I. Kontogounis, and P.A. Mitkas. Agent mertacor: A robust design for dealing with uncertainty and variation in scm environments. *Expert Systems with Applications*, 35(3):591–603, 2008.
3. J. Collins, R. Arunachalam, N. Sadeh, J. Eriksson, N. Finne, and S. Janson. The supply chain management game for the 2007 trading agent competition. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, 2006.
4. C. de Snoo, W. van Wezel, J.C. Wortmann, and G.J.C. Gaalman. Coordination activities of human planners during rescheduling: Case analysis and event handling procedure. *International Journal of Production Research*, To appear, 2010.
5. K. Främling, M. Harrison, J. Brusey, and J. Petrow. Requirements on unique identifiers for managing product lifecycle information - comparison of alternative approaches. *International Journal of Computer Integrated Manufacturing*, 20(7):715–726, 2007.
6. K. Främling, J. Holmström, T. Ala-Risku, and M. Kärkkäinen. Product agents for handling information about physical objects. Technical report, Helsinki University of Technology, 2003.
7. E. Huvio, J. Grönvall, and K. Främling. Tracking and tracing parcels using a distributed computing approach. In *Proc of NOFOMA'02*, pages 29–43, 2002.
8. P.R. Jordan, C. Kiekintveld, J. Miller, and M.P. Wellman. Market efficiency, sales competition, and the bullwhip effect in the tac scm tournaments. In *Agent-Mediated Electronic Commerce*, volume 4452 of *Lecture Notes in Computer Science*, pages 62–74. Springer, 2007.
9. C. Kiekintveld, J. Miller, P.R. Jordan, and M.P. Wellman. Controlling a supply chain agent using value-based decomposition. In *Proc. of the 7th ACM conference on Electronic commerce*, pages 208–217, 2006.
10. M. Kärkkäinen, J. Holmström, K. Främling, and K. Artto. Intelligent products - a step towards a more effective project delivery chain. *Computers in Industry*, 50(2):141–151, 2003.
11. P. McCullen and D. Towill. Diagnosis and reduction of bullwhip in supply chains. *Supply Chain Management: An International Journal*, 7(3):164–179, 2002.
12. D. McFarlane, S. Sarma, J.L. Chirn C.Y. Wong, and K. Ashton. Auto id systems and intelligent manufacturing control. *Engineering Applications of Artificial Intelligence*, 16(4):365–376, 2003.

13. G.G. Meyer, K. Främling, and J. Holmström. Intelligent products: A survey. *Computers in Industry*, 60(3):137–148, 2009.
14. J. Orlicky and G.W. Plossl. *Orlicky's material requirements planning*. McGraw-Hill, 1994.
15. D. Pardoe and P. Stone. Tactex-03: A supply chain management agent. *SIGecom Exchanges*, 4(3):19–28, 2004.
16. D. Pardoe, P. Stone, and M. VanMiddlesworth. Tactex-05: An adaptive agent for tac scm. In *Agent-Mediated Electronic Commerce*, volume 4452 of *Lecture Notes in Computer Science*, pages 46–61. Springer, 2007.
17. S. Sarma, D.L. Brock, and K. Ashton. The networked physical world. Technical Report MIT-AUTOID-WH-001, MIT Auto-ID Center, October 2000.
18. M. Stan, B. Stan, and A.M. Florea. A dynamic strategy agent for supply chain management. In *Proc. of SYNASC'06*, pages 227–232, 2006.
19. M. Strassner and T. Schoch. Todays impact of ubiquitous computing on business processes. In *Proc. of Pervasive'02*, pages 62–74, 2002.
20. P. Toulis, D. Kehagias, and P.A. Mitkas. Mertacor: a successful autonomous trading agent. In *Proc. of AAMAS'06*, pages 1191–1198, 2006.
21. C.Y. Wong, D. McFarlane, A.A. Zaharudin, and V. Agarwal. The intelligent product driven supply chain. In *Proc. of SMC'02*, 2002.

Appendix: Detailed simulation results

	$n = 0$		$n = 5$		$n = 10$	
Agent	μ	σ	μ	σ	μ	σ
GRUNN	99.43 %	0.77 %	99.55 %	0.36 %	99.45%	0.34 %
TacTex-07	99.50 %	0.51 %	98.20 %	1.35 %	97.80%	1.50 %
PhantAgent-07	99.72 %	0.24 %	88.73 %	2.85 %	75.94 %	5.71 %
DeepMaize-07	98.62 %	1.09 %	97.00 %	2.06 %	96.01 %	3.79 %
Mertacor-08	95.25 %	3.98 %	90.81 %	7.86 %	89.85 %	8.04 %
Dummy	42.83 %	24.35 %	26.14 %	8.95 %	23.04 %	12.18 %

Table 1. Percentage of orders finished in time

	$n = 0$		$n = 5$		$n = 10$	
Agent	μ	σ	μ	σ	μ	σ
GRUNN	-$5.528	$6.479	-$8.028	$3.969	-$15.532	$3.901
TacTex-07	$16.093	$7.859	$13.405	$3.723	$5.691	$5.244
PhantAgent-07	$16.588	$7.099	$8.198	$4.846	-$6.904	$4.220
DeepMaize-07	$11.579	$5.971	$9.336	$3.359	$3.550	$4.714
Mertacor-08	$6.010	$6.070	$3.764	$4.745	-$3.675	$5.272
Dummy	-$10.210	$15.066	-$23.110	$26.488	-$21.562	$17.139

Table 2. Profit in million dollars

	$n = 0$		$n = 5$		$n = 10$	
Agent	μ	σ	μ	σ	μ	σ
GRUNN	$309	$90	$278	$66	$242	$87
TacTex-07	$347	$69	$384	$52	$326	$66
PhantAgent-07	$255	$58	$193	$25	$155	$28
DeepMaize-07	$235	$39	$254	$42	$228	$40
Mertacor-08	$256	$61	$263	$33	$220	$46
Dummy	$250	$95	$247	$85	$208	$88

Table 3. Storage costs per accepted order in dollars