Smart Objects System: A generic system for enhancing operational control

Gerben G. Meyer, W.H. (Wilrik) Mook, and Men-Shen Tsai

Abstract Many companies are making considerable investments in tracking technology, such as GPS and RFID. Although tracking technology captures vast amounts of information about the ongoing operations, companies struggle to effectively apply this captured information for enhancing their operational control. In order to contribute in solving this problem, this paper presents a generic system for enhancing operational control, which applies the captured information in a more effective way. The proposed system is based on the approach of intelligent products. The intelligent products represent physical objects, and are capable of autonomously performing some of the repetitive tasks required for operational control. The usefulness of the system is demonstrated by presenting the results of several applications of the system.

1 Introduction

Many companies are making considerable investments in tracking technology, such as GPS and RFID, which is capable of capturing vast amounts of real-time information about the on-going operations (see e.g. [1, 24]). These investments are typically made with the aim to enhance the control of the ongoing operations, as the captured information can be applied for detecting unexpected events as well as for directing the decision making aimed at mitigating the consequences of these unexpected events [3]. Examples of

Gerben G. Meyer
National Taipei University of Technology, Taiwan, e-mail: gerben.meyer@gmail.com

W.H. (Wilrik) Mook
University of Groningen, The Netherlands, e-mail: w.h.mook@rug.nl

Men-Shen Tsai
National Taipei University of Technology, Taiwan, e-mail: mstsai@ntut.edu.tw

unexpected events include delays in transportation and deliveries, failing resources, and products which are missing or malfunctioning. Hence, operational control is required for determining whether the operations are being performed as intended, and to make short-term adjustments for mitigating the negative impact of unexpected events on the ongoing operations [25].

Many companies however struggle to effectively apply the captured information for enhancing their operational control. Regarding this issue, Crainic et al. [5] state that "*The information is there. One only needs the appropriate methodology to transform these data into accurate and timely decisions.*". Many authors confirm this observation of Crainic et al., showing that computer-based support systems often fail to provide the required information and functionality to support planners responsible for the control of ongoing operations (see e.g. [2, 6, 11, 13, 23]). Therefore, operational control is still typically a manual task, leaving most of the captured information unused.

The goal of the research as described in this paper is to develop a generic system for enhancing operational control by applying the captured information in a more effective way. A design science approach [8, 21] has been adopted for the development and evaluation of this system. The proposed system as presented in this paper is based on the approach of intelligent products [16]. Intelligent products can represent individual physical objects such as products and resources in the system, and are capable of autonomously performing some of the repetitive tasks required for operational control. Such tasks include analyzing the captured information on a low level of granularity, presenting the captured information in a more comprehensive way, and providing notifications when unexpected events occur. In this way, the proposed system enables companies to enhance their operational control by applying the information captured by tracking technologies in a more effective way.

The development of the system reported here has been supported by extensive research related to the concept of intelligent products, which comprises two main sources of knowledge. The first source is formed by state of the art theories and frameworks in the field of intelligent products (see e.g. [10, 12, 26]). The recent research trends, emerging architectural designs, and novel application domains in this field are surveyed in [16], leading to the formulation of a classification model which has been used as a research framework. The second source of knowledge is based on application oriented experience, which was gained during the iterative development of an intelligent products architecture and the evaluation of that architecture by means of validating various prototypes [14].

The remainder of this paper is structured as follows. Next, a short analysis is presented, leading to the formulation of system requirements. Afterwards, Section 3 presents the proposed generic system for enhancing operational control. In order to demonstrate its usefulness, Section 4 presents the results of several applications of this system. Conclusions are provided in the last section.

2 Requirements

As mentioned in the introduction, operational control is typically still a manual task, leaving most of the captured information about the ongoing operations unused. This is due to the fact that current computer-based support systems often fail to provide the required information and functionality to support planners responsible with this task. Hence, new system requirements are needed, in order to develop a system which is able to contribute in solving this problem.

Tracking technology typically captures vast amounts of information about the ongoing operation. For example, fleet management systems capture every movement of every vehicle in a company's fleet (see e.g. [9]), manufacturing control systems capture every operation performed by a company's manufacturing resources (see e.g. [12]), and product lifecycle management systems capture every action performed on every individual product (see e.g. [7, 22]). In order to apply this captured information in a more effective way, the system should first of all be able to access all this information.

- *Requirement #1*: The system should be able to access all captured information.

Presenting all captured information to the planners will not improve the operational control of the ongoing operations, as the amount of captured information is likely to overwhelm the planners. Hence, the system should only present the information which the planners require for performing operational control.

- *Requirement #2*: The system should only present the information relevant for performing operational control.

In order to effectively perform operational control, it is key that unexpected events are detected in time. An early detection of an unexpected event gives the planners more time to assess the situation at hand and to determine which short-term adjustments are needed for mitigating the consequences of the unexpected event. Therefore, it is important that the system is able to pro-actively notify the planners when unexpected events occur.

- *Requirement #3*: The system should pro-actively provide notifications when unexpected events occur.

For planners, manually finding a suitable control decision for mitigating the consequences of an unexpected event can be troublesome, due to the complexity of the operations under control and the time available for making such a decision. Hence, it is beneficial if the system is able propose potential control decisions, allowing the planners to choose the most appropriate decision.

- *Requirement #4*: The system should discover and propose potential control decisions.

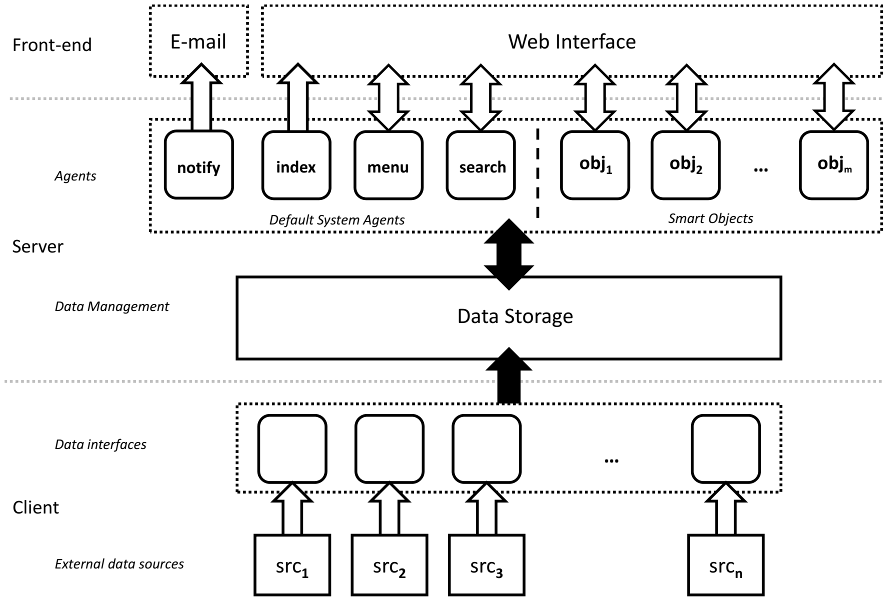

Fig. 1 System architecture

3 System design

This section describes the Smart Objects System (SOS), the proposed generic system for enhancing operational control. First, the overall system architecture is presented. Afterwards, the structure and the behavior of the proposed system are described.[1]

3.1 Architecture

The overall architecture of SOS is shown in Figure 1. As can be seen from the figure, an application built with SOS needs information about physical objects from one or more external data sources. For each of these external data sources, a data interface needs to be provided, which interprets the external data and converts the structure of this data into a structure compatible with the SOS data storage. For every physical object of which information is stored in the data storage, the server creates an agent which can add intelligence to this physical object. In this way, a so called *smart object* is created. Every agent can execute its own application specific behavior, for example

[1] A more extensive description of the system can be found on the SOS project page: `http://code.google.com/p/smart-objects-system/`

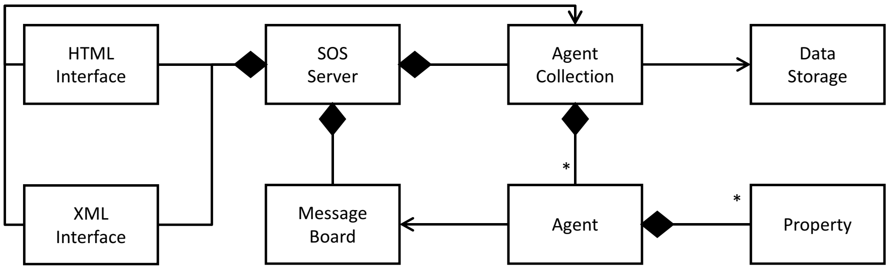

Fig. 2 System structure

to determine which information will be displayed in the web interface. Besides the agents representing physical objects, several default system agents are always present. The *index*, *menu*, and *search*-agent are responsible for generating the generic parts of the web interface, and the *notify*-agent is responsible for generating e-mail notifications.

3.2 Structure

The basic structure of the system is shown in Figure 2, which shows a simplified UML class diagram containing the most important classes of the system. The purpose of each of these classes will be shortly explained next.

- The *SOSServer* class is the starting point of the system. It starts the HTTP and XML interfaces, prepares the AgentCollection and MessageBoard, and initializes the default system agents.
- The *Agent* class is an abstract class, which acts as a base for all application specific agents. Every agent represents one physical object in the system, of which the information is maintained through a set of properties. Methods for getting and setting these properties, as well as learning its current status and executing application specific behavior are defined in this class.
- A *Property* is a basic data structure which defines a single property of an agent. A number of different property types have been defined, such as text, number, time, and location.
- The *AgentCollection* is used for managing all agents which are currently present in the system, and the *DataStorage* is used for storing all the properties of agents in a database.
- The *MessageBoard* is used for communication between agents.
- The *HTMLInterface* provides the web interface for system users, and the *XMLInterface* provides the interface through which external data about physical objects can be added to the data storage.

3.3 Behavior

The behavior of the agents is largely application specific, and therefore different for every application developed with SOS. This section describes on a more generic level how the behavior of the agents can be used to meet the requirements introduced in Section 2 when developing an application with SOS. This behavior is introduced according to the three levels of intelligence for intelligent products as prescribed by Meyer et al. [16]: information handling, problem detection, and decision support.

3.3.1 Information handling

When an application is developed with SOS, it is key to determine which physical objects will be represented by agents in the system. Information about these objects captured by tracking technology or other information systems has to be made available to the agents representing them. For this purpose, application specific data interfaces have to be developed, which translate the external data into data understandable by the agents. In that way, *Requirement #1* will be met, and every agent is enabled to collect all available information related to the physical object it is representing.

Presenting all available information to the system users will not enhance operational control. As every agent has to determine which information it will present to the system users (see Figure 1), the agents representing physical objects have to be developed in such a way, that only information important for performing operational control in the specific application is displayed. In that way, *Requirement #2* will be met.

3.3.2 Problem detection

By means of the web interface, SOS enables the system users to train the agents, as the system users can inform them whether their current situation is problematic or not. The trained agent will generate a training instance based on the provided status and the information available on the physical object it is representing. This training instance will be stored in the data storage and will be shared among other agents representing the same type of physical object. By using a machine learning classifier [20], the agents are enabled to continuously determine their status. In order to meet *Requirement #3*, the system users will be directly notified by means of an e-mail message when the status of an agent suddenly becomes problematic. Moreover, when requested by a system user, the agents will also provide this information through the web interface.

3.3.3 Decision support

Although no generic behavior for agents is provided on this level of intelligence, every agent developed for SOS can perform application specific behavior needed for decision support. Agents can for example communicate and negotiate with each other by using the message board, in order to discover potential control decisions. By developing such application specific agent behavior, *Requirement #4* can be met, as one of the applications presented next demonstrates.

4 Applications

Two applications of SOS have been developed, in order to demonstrate the usefulness of the proposed system. These applications will be shortly discussed next.

4.1 Transportation

One application developed with SOS is an application for enhancing operational control of a medium-sized road freight transportation company. The planners of this company struggle to detect unexpected events in time, as the amount of information collected by the tracking technology in place is too high to be analyzed manually. Despite the available tracking technology, the planners are only informed about unexpected events through conventional methods, such as phone calls with customers and on-route truck drivers, often resulting in problems being detected too late.

Within the application developed with SOS, every truck of the company as well as every pallet to be transported by the company is represented by an agent. This application is focused on the behavior of these agents on the level of information handling and problem detection, as the main goal of the application is to assist the planners in handling the available information and detecting unexpected events in time. Hence, on the level of information handling, every agent collects all information available in the existing information systems of the company related to the physical object it represents and provides the planners with a comprehensive overview of the plan it is involved in combined with the actual progress of this plan. On the level of problem detection, pallet agents are trained to have a problematic status when their expected delay is one hour or more, and truck agents are trained to have a problematic status when they are transporting one or more pallets with a problematic status. The web interface of the application enables the planners to directly determine which agents have a problematic status. Moreover, e-mail messages are send automatically to the planners to pro-actively inform them when the status of an agent suddenly becomes problematic.

The developed system has been used for conducting a pilot study at the aforementioned transportation company. This study showed that the system was able to correctly detect problems caused by unexpected events. The planners were also correctly notified about problems which were not yet observed by themselves, but nevertheless required immediate control decisions. Moreover, the pilot study showed that the information presented by the agents enabled a better understanding of the unexpected events including their impact on the ongoing operations. More details on the developed application as well as the performed evaluation can be found in [15] and [17].

4.2 Production

Another application developed with SOS is an application for performing operational control of a manufacturer of personal computers within the Trading Agent Competition Supply Chain Management (TAC SCM) simulated supply chain [4]. Within the TAC SCM simulated supply chain, every manufacturer has to procure customer orders, buy computer components, assemble computers, and ship the computers to the customers. For this purpose, the manufacturer has a computer factory containing an assembly cell capable of assembling any type of computer, and a warehouse that stores both components and assembled computers.

Within the TAC SCM manufacturer developed with SOS, every product ordered by a customer is represented by an agent. Such an agent is responsible for the complete processing of one final product. Hence, an agent is responsible for acquiring the components required for assembling the product, acquiring the required production capacity, as well as arranging the shipment of the assembled product to the customer who ordered it. The developed application is focused on the behavior of these agents on the level of decision making, as information handling and problem detection are less important in a simulated environment with no involvement of human planners. Hence, on the level of decision making, every product agent negotiates with other agents to plan and control the distribution of components, production capacity, and shipping capacity. For each of these tasks, an auctioning approach is used, in which the agents can bid on the required components and resources.

The results of the conducted simulations showed that the developed TAC SCM manufacturer is very robust in terms of handling unexpected events. Although the manufacturer had to deal with late supplier deliveries as well as broken components, it was still able to finish nearly all of the requested products in time, without increasing the component inventory “safety stock”. More details on the developed application as well as the simulation results can be found in [18] and [19].

5 Conclusions

Many companies struggle to effectively apply the vast amounts of information captured by tracking technology for enhancing their operational control. Therefore, a generic system for enhancing operational control has been developed, with the goal to apply the captured information in a more effective way. By adopting the intelligent products approach, the proposed system is capable of autonomously performing some of the repetitive tasks required for operational control. The presented applications have demonstrated that the proposed system contributes to enhancing operational control by applying the captured information in a more effective way.

Two managerial implications can be gleaned from the research as presented in this paper. Firstly, investments of companies in tracking technology are not likely to directly enhance operational control. Secondly, a system based on intelligent products as presented in this paper can be applied to overcome this problem. Due to the generic nature of this system, other companies facing similar problems when applying captured information for enhancing operational control are likely to benefit from the system as well. Accordingly, future work will be focused on developing additional applications for companies facing similar problems. These applications can contribute to confirming as well as generalizing the results and insights as presented in this paper.

Acknowledgments

The authors would like to express their gratitude to Gijs Roest for his assistance in developing the Smart Objects System. This research is supported by the National Energy Project of the National Science Council, Taiwan, under grant number NSC 101-3113-P-006-020.

References

1. R. Angeles. Rfid technologies: Supply-chain applications and implementation issues. *Information Systems Management*, 22(1):51–65, 2005.
2. A. Budihardjo. *Planners in action: roadmap for success: an empirical study on the relationship between job decision latitude, responsiveness and planning effectiveness in road transport companies.* PhD thesis, University of Groningen, Groningen, The Netherlands, 2002.
3. P. Buijs, N.B. Szirbik, G.G. Meyer, and J.C. Wortmann. Situation awareness for improved operational control in cross docking: An illustrative case study. In *Proceedings of the 14th IFAC Symposium on Information Control Problems in Manufacturing*, 2012.
4. 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 CMU-ISRI-07-10, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 2006.
5. T.G. Crainic, M. Gendreau, and J.-Y. Potvin. Intelligent freight-transportation systems: Assessment and the contribution of operations research. *Transportation Research Part C: Emerging Technologies*, 17(6):541–557, 2009.
6. C. de Snoo, W. van Wezel, and R.J. Jorna. An empirical investigation of scheduling performance criteria. *Journal of Operations Management*, 29(3):181–193, 2011.
7. K. Främling, T. Ala-Risku, M. Kärkkäinen, and J. Holmström. Agent-based model for managing composite product information. *Computers in Industry*, 57(1):72–81, 2006.
8. A.R. Hevner, S.T. March, J. Park, and S. Ram. Design science in information systems research. *MIS Quarterly*, 28:75–105, 2004.
9. S. Ichoua, M. Gendreau, and J.-Y. Potvin. *Dynamic fleet management: concepts, systems, algorithms & case studies*, chapter Planned route optimization for real-time vehicle routing, pages 1–18. Springer, 2007.
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. B.L. MacCarthy and J.R. Wilson. *Human performance in planning and scheduling*, chapter Influencing industrial practice in planning, scheduling and control, pages 451–461. Taylor & Francis, 2001.
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. K.N. McKay and V.C.S. Wiers. *Practical production control: a survival guide for planners and schedulers*. J. Ross Publishing, 2004.
14. G.G. Meyer. *Effective Monitoring and Control with Intelligent Products*. PhD thesis, University of Groningen, 2011.
15. G.G. Meyer. *Effective Monitoring and Control with Intelligent Products*, chapter System Prototype for Transportation, pages 111–139. University of Groningen, 2011.
16. G.G. Meyer, K. Främling, and J. Holmström. Intelligent products: A survey. *Computers in Industry*, 60(3):137–148, 2009.
17. G.G. Meyer, G.B. Roest, and N.B. Szirbik. Intelligent products for monitoring and control of road-based logistics. In *Proceedings of the 2010 IEEE International Conference on Management and Service Science*, Wuhan, China, August 2010.
18. G.G. Meyer and J.C. Wortmann. Robust planning and control using intelligent products. In *Agent-Mediated Electronic Commerce*, volume 59 of *Lecture Notes in Business Information Processing*, pages 163–177. Springer-Verlag, 2010.
19. G.G. Meyer, J.C. Wortmann, and N.B. Szirbik. Production monitoring and control with intelligent products. *International Journal of Production Research*, 49(5):1303–1317, 2011.
20. T.M. Mitchell. *Machine learning*. McGraw-Hill, 1997.
21. K. Peffers, T. Tuunanen, M.A. Rothenberger, and S. Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77, 2007.
22. M. Rönkkö, M. Kärkkäinen, and J. Holmström. Benefits of an item-centric enterprise-data model in logistics services: A case study. *Computers in Industry*, 58(8-9):814–822, 2007.
23. A. Rushton, J. Oxley, and P. Croucher. *The handbook of logistics and distribution management*. Kogan Page, 2000.
24. J. Schumacher and K. Feurstein, editors. *Proceedings of the 3rd European conference on ICT for Transport Logistics*, Bremen, Germany, November 2010.
25. N. Slack, S. Chambers, and R. Johnston. *Operations management*. Pearson Education, 2004.
26. P. Valckenaers, B. Saint Germain, P. Verstraete, J. Van Belle, Hadeli, and H. Van Brussel. Intelligent products: Agere versus essere. *Computers in Industry*, 60(3):217–228, 2009.