ProVis.Agent: An Agent-Based Production Monitoring And Control System

B. Jacke[a], O. Sauer[b], G. Sutschet[b]

[a] *DaimlerChrysler AG, Bremen, Germany*
[b] *Fraunhofer Institute for Information and Data Processing, Karlsruhe, Germany*

Abstract

ProVis.Agent is the first agent-based production monitoring & control system for distributed real-time production monitoring. Its functionality is based on *ProVis.NT*, the proven object-oriented control system which monitors and controls e.g. the body, paint and assembly shops in DaimlerChrysler's automotive plant in Bremen, Germany. Forming an integral part of the manufacturing execution system (MES), *ProVis.Agent* for the first time allows to be integrated with other shop-floor related applications, such as body identification, worker information, sequence setup, etc.

In this paper we show a case study of implementing agent-based technology to a classical field of automation.

1. Introduction

In today's automotive plants we find heterogeneous software systems for different types of tasks both for factory planning and the manufacturing operations. IT-systems used for factory planning are summarized as 'digital factory tools'. On the operating level software systems are not yet integrated and thus they support separate tasks such as production order control, production monitoring, sequence planning, vehicle identification, quality management, maintenance management, material control and others.

Production monitoring and control systems (PMC) play a central role to the classical automation field. The main function of those systems is to gather signals produced by plants and programmable logic controllers (PLCs), combine them to control relevant contexts, visualize them and provide facilities to operate them.

While visualization and operation of process signals and contexts are classic functions of so called SCADA systems[1], the main work of real time signal processing and interfacing to production plants is done by PMCs [1]. Today these PMCs are usually implemented as object oriented systems, interfacing with their environment by standardized protocols (e.g. OPC[2] [2]).

The Fraunhofer Institute for Information and Data Processing (Fraunhofer IITB) has an almost 30-years-experience in developing novel PMCs especially in the fields of automotive plants and steel production. In 2005 DaimlerChrysler Bremen ordered a new generation of PMC for the coming C-type car, starting its production at the beginning of 2007.

[1] SCADA: Supervisory Control And Data Acquisition
[2] OPC: OLE for Process Control

2. Motivation

Facing the fact that PMCs usually are operated in highly distributed hard- and software environments, where a continuous connection of all components involved can not be guaranteed, the implementation of each subsystem needs to be able to operate autonomously at least for a certain span of time.

According to the specific needs of a special plant PMCs must be able to either follow a central, decentral or mixed control approach. Whereas e.g. DaimlerChrysler's automotive plant in Bremen implements one central control room for each of the assembly sections bodyshell work, paint shop and final assembly (see Fig 1), other automotive plants follow a decentralized structure, where visualization and operation are performed directly from the operation stations situated on the shop floor.

Fig. 1. Control room of DaimlerChrysler's automotive plant in Bremen, Germany

Independency from special operating systems becomes more and more important, because the traditional Microsoft Windows applications in future have to be connected to Linux systems and other vendor specific operating systems.

Furthermore the integration of "neighboring" IT-systems[3], such as quality management, maintenance and repair, sequence scheduling, car body identification etc., should be made as easy as possible. A connection between these applications lead to both better information to react on unexpected disturbances on the shop floor and as well as to higher transparency concerning production related information. Difficulties arise if these MES systems come from different vendors. Up to now there is no standardized way of communication for such systems leading to problems of misunderstandings about concepts to communicate.

The engineering environments of all concurrently running systems are usually specific to the single systems, i.e. there is no common repository for plant specifications, signal types, signal wiring, etc. A large part of the information needed to operate the single systems has to be provided redundantly to the engineering parts of the subsystems. Besides the fact that this leads to an unnecessary engineering overhead, this is a possible source for errors due to manual data input, especially in cases of reconfiguration of the plant.

In the years to come many of the assembly lines will be constructed of equipment and PLCs which are self-aware of the functions they can perform and they way the can be parameterized (engineered). Some of today's assembly robots e.g. are delivered with web servers allowing to visualize their state and reconfigure them. Assuming that this is a trend for the future, the possibility of 'Plug & Produce' assembly components becomes more and more likely to revolutionize the engineering of production systems. This raises the need for a standardized way of communication among different engineering systems.

Fraunhofer IITB's business unit dealing with production monitoring systems is already working on intelligent engineering tools supporting the 'plug & produce' approach.

3. Challenges

For production monitoring and control systems have to be able to visualize and operate the state of production plants in real time, there is a need for very fast communications of signals and operation actions.

Nevertheless one has to keep in mind that a connection via usual bus systems is not guaranteed to be possible at any time during the production process. Therefore special autonomous features have to be implemented to assure a correct function of the plant.

For certain events (e.g. shift changeover) there is a need to handle a tremendous amount of data and signals in very short time. Several hundred signals per second have to be communicated, gathered, sorted and combined and handed over to subsequent systems like databases for evaluation and evaluation purposes.

A PMC has furthermore to provide several different protocols[4] to underlying production lines,

[3] usually subsumed as Manufacturing Execution Systems (MES)

[4] Such as OPC, MMS, simple TCP-IP and others

which implement different abilities, different formats, different frequencies of communication and different data structures. Having an overall model for process states a PMC has to integrate all this information in real time.

4. Integration aspects

Given that the development of PMCs can not be regarded as stand-alone systems, it has to take into account that there must be many interfaces to related IT-systems. These systems have usually evolved "historically" in the factories and can – for economical reasons – not simply be replaced by new systems, which could be integrated more easily.

Therefore an integration platform for PMC systems with other MES systems is needed. Whereas several vendors provide a platform for their own MES components (e.g. SIEMENS, Wonderware, Rockwell Automation, etc.), it is typically not possible to operate systems of different vendors on the same platform. Dedicated interfaces have to be provided for each specific configuration of MES components coming from different vendors.

Here is a challenging need for standardization. In principle there are several ways of integrating concurrent systems which all have several advantages and disadvantages. Three possible options will be outlined in the following:

4.1. Vendor proprietary platform

Currently the stat of the art for integration of MES systems is to use only MES components from one vendor. These usually operate via a proprietary integration platform specific for each vendor. Examples for this kind of platforms are the "Industrial Framework" of SIEMENS or Wonderware's Archestra platform.

The main structure of this architecture is shown in Fig. 2. All MES modules reside inside a uniform framework with limited access from outside the framework. The framework itself is designed to optimize interaction of especially the MES modules of this vendor.

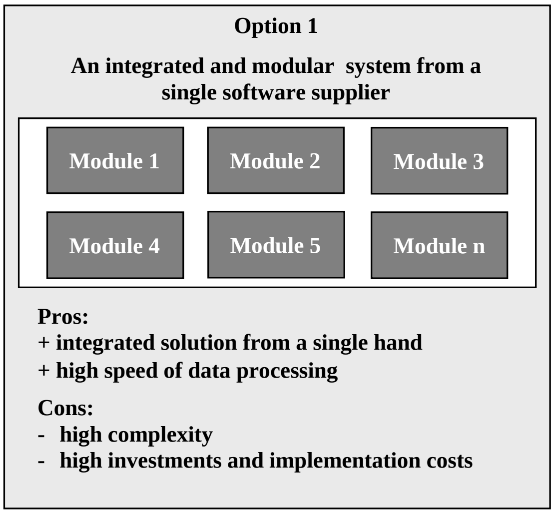

Fig. 2. MES integration platform of a single supplier

One advantage of these platforms is, that the complete solution is delivered from one vendor being in charge of the correctness of the whole integrated system. Usually the optimization of communication between the in-house components leads to a very high speed of data processing.

On the other hand these highly complex frameworks are usually not well suited to integrate third party products. Higher investment for interface implementation is needed whenever a new MES configuration is to be implemented.

4.2. Database centered approach

In contrast to vendor proprietary platforms and due to the mentioned disadvantages today the integration aspect is in most cases supported by databases (see Fig 3). It must be added that a couple of the above-mentioned systems, particularly those for production monitoring, worker information, object identification, etc., require real-time data processing instead of database solutions because they receive a large number of signals from the production equipments' PLCs. For real-time applications, database solutions are often not fast enough.

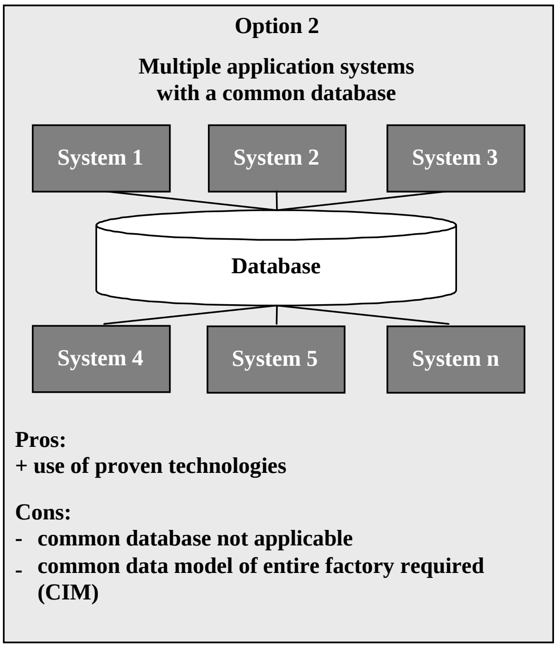

Fig. 3. Database centered approach

Another disadvantage of integration by means of a database is the required data model for all related applications. This data model is comparatively inflexible, especially if changes or extensions of functionalities are required or if new IT systems must be added.

In the late 1990s a team of the Fraunhofer IPK developed a first concept for an Asian OEM (Hyundai Motor Company) allowing to integrate some of the above-mentioned IT systems, particularly those on the cell manufacturing level (see Fig 4). Being ahead of their time, the Fraunhofer people designed a cell manufacturing system that combined production monitoring, quality control, maintenance / repair and tracking information for each shop - press, body, paint and trim shop. Applying the available software technology of that time, the plant in question went into operation in 1998.

4.3. Integration on agent platform

Today it is evident that new software technologies have to be used to allow for a genuine integration of IT systems for production equipment, quality issues, provided parts and shift output and to preserve the existing software functionalities.

A more promising technology for integrating existing software systems and their functionalities and to add assistant systems for the shop floor staff is to be found in software agents [4]. In the academic field, agent technologies have a long tradition, but their use in production and real-time applications has been very limited yet. Describing the state-of-the-art for agent-based systems we concentrate in the following on practical applications, esp. in automation applications.

One of the first promising applications is described

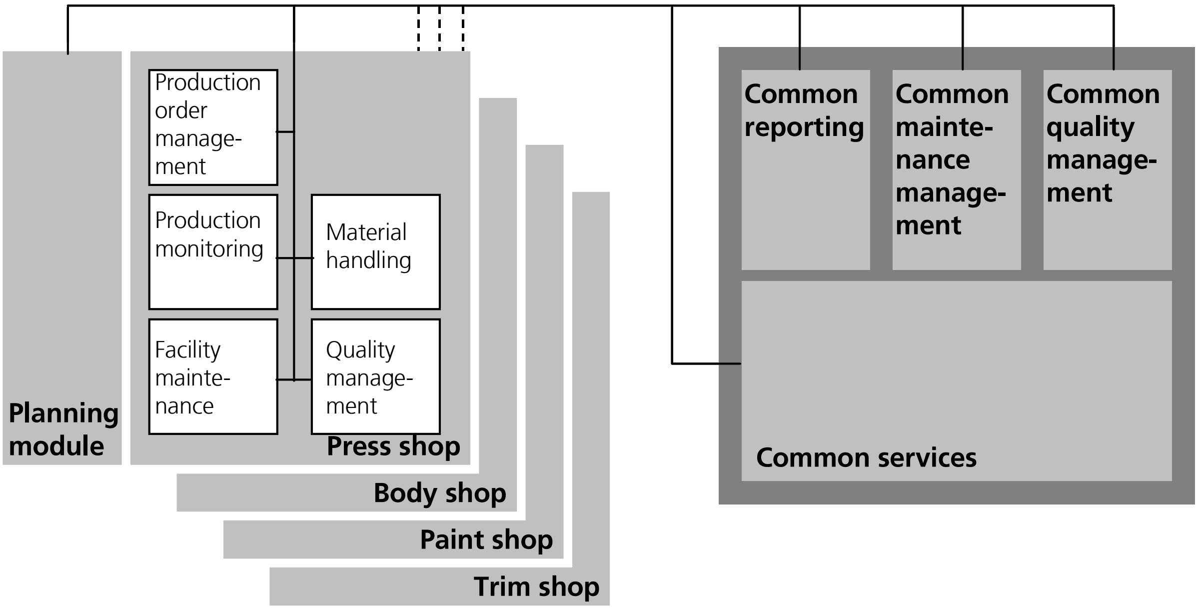

Fig. 4. Overview of the cell manufacturing concept developed by Fraunhofer IPK [3]

by authors from DaimlerChrysler [5, 6] who have implemented a manufacturing cell for cylinder heads and other engine parts that is completely controlled by a software-agent-based system.

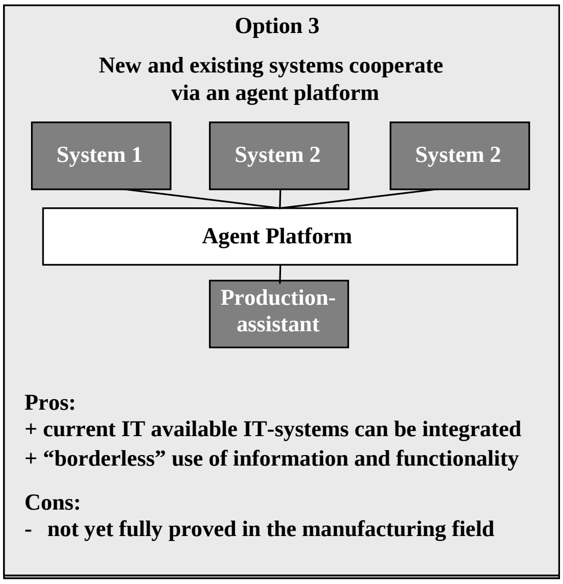

Fig. 5. Integration on an agent platform

The manufacturing cell works in serial production and has produced excellent results concerning capacity utilization and smoothed production. The control logic developed in this project by the PLC-supplier [7] is offered for further industrial applications. Most authors refer to this example to prove agent-technology being applicable on the shop floor.

Hertzberg et. al. describe results of the AgenTec project which have been applied to a prototype of a commissioning system to illustrate and test the integration of heterogeneous software and control systems [8]. Further examples for potential application in automation technologies are summarized by Urbano et. al. [9]. However, a convincing industrial application is still missing. Bruccoleri et. al. have stated, that agent based ERP approaches are currently being developed for application within FIAT [10].

Agent platforms usually offer a standard way of communication among the agents hosted on the platform. Using ontologies for different subsystems assures the uniqueness of concepts to be exchanged.

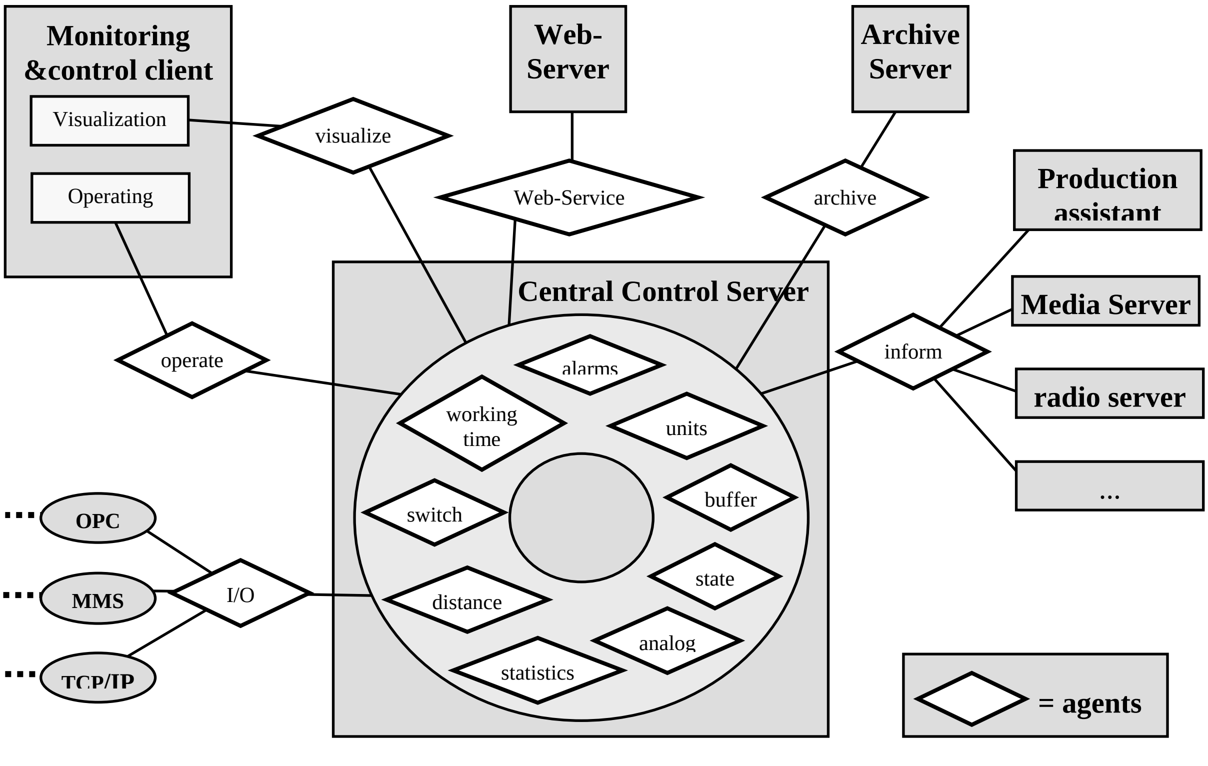

Fig. 6. Agent structure of *ProVis.Agent*

5. Solution

Starting from the proven object oriented PMC *ProVis.NT* Fraunhofer IITB has decided to use software agents for the means of integration when developing further PMCs or other MES components.

Fig 6 shows the use of software agents within the actual system architecture of *ProVis.Agent*. The central Monitoring & Control Server consists of a collection of cooperating software agents. Each of these agents covers one piece of functionality already contained in the former system *ProVis.NT*. It contains the functional treatment of different types of signals (e.g. switches, analog values, distances, etc.) as well as working time models, alarming and statistical data.

The I/O-agent encapsulates different types of I/O-channels (OPC, MMS-Light and pure TCP/IP) and allows the Control Room Server to have a uniform look on all the signals delivered by underlying systems. It is essential that this agent implements interfaces to different communication standards, because in industrial applications a PMC has to connect to new plants – providing modern protocols – as well as "historically grown" plants supporting only rudimentary communication facilities.

A visualization agent is used for interfacing with a variety of commonly used SCADA systems (WinCC, FactoryLink, etc.) as well as with the new real-time visualization tool *ProVis.Visu*, which has also been developed by Fraunhofer IITB.

The operating agent always provides the operating context for single signals or complex actions. This context may either be only the name and operating mode of a single underlying plant, but it may also be a combination of several signals of different types coming from several sub-plants. The main objective of the operating agent is to provide the operator with all information needed to perform even complex operations (such as changing the working time model for different plants). At any time the operator must be in a position to evaluate the consequences of his actions correctly.

A variety of special agents is used for interfacing to existing subsystems The Web-Server agent interfaces the PLC to a statistical analysis system, which can either be the in-house product *Provis.Paula* or a third party software. The Archive-Server agent reports production relevant information to a archive database, where it is stored and condensed for quality checks and future planning.

The information agent finally handles the connections to the Media Server and the radio server as well as to special assistance components.

All agents contained in *ProVis.Agent* communicate via an ontology based communication schema. The ontology is open for neighboring systems to access any of the data contained in *ProVis.Agent*. Special care is taken to ensure the correctness of information to be

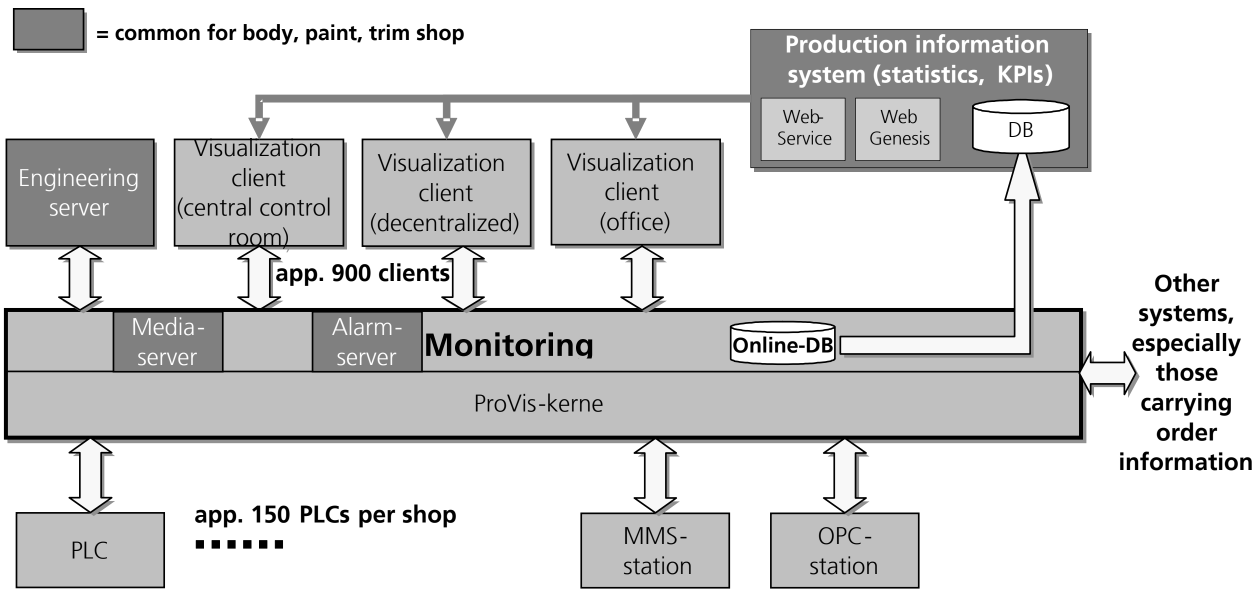

Fig. 7. Actual architecture of agent based PMC

changed by outer systems.

Fraunhofer decided to base ProVis.Agent on the FIPA[5] [11] standard. All of the agents have been implemented on top of the agent platform JADE[6] [12]. The communication ontology has been developed using Protégé[7] [13].

An example of the architecture of such a new production monitoring and control system which is now open for connection to IT systems related to logistics, quality management or building utilities control is shown in Fig 7.

6. Conclusion and Outlook

ProVis.Agent has been implemented and is currently in the test phase. It will be delivered to DaimlerChrysler in April 2006 and start operation with the start up of new facilities for the new C-class model.

Let's assume it is useful to bring some of the existing operational IT systems of an automotive plant together - the user would be lost in view of the variety and abundance of functionalities. This means that a user-oriented assistant software that makes available the required functions and information for a specific working activity is essential for the users on the shop floor to take the required and expected decisions.

For both, to prove the agent technology concept and to demonstrate such an assistant systems Fraunhofer IITB has developed a real time simulation system that calculates for the 3 – 5 coming shifts the production output regarding all actual production breakdowns indicated by the production monitoring system as well as the buffer status calculated according to the current production output (Figure 7). In this case the interface shown in Fig. 8 is WinCC.

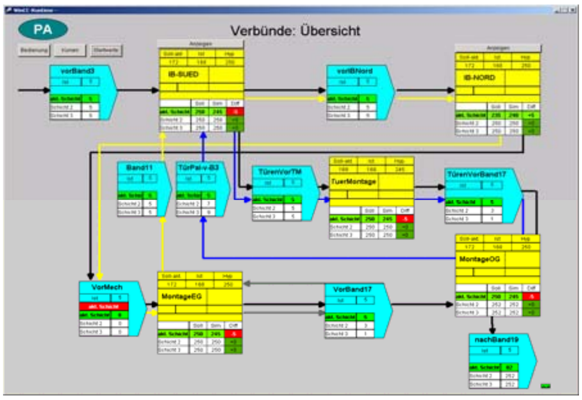

Fig. 8. Short-time simulation system as production assistant tool connected to the production monitoring system

With the help of this assistant software, which is connected to the monitoring systems by a software agent, the shop floor people can react to failures on the shop floor level quickly and correctly. They are now able to adjust line speed and the work force for the coming shifts and to take efficient measures to keep the output as high as possible.

Acknowledgements

The authors like to thank the DaimlerChrysler unit in the Bremen plant that is in charge of planning production monitoring systems (FP SNT 5). Without their innovation related mind the realization of ProVis.Agent would not have been possible.

References

[1] Rao B. K. N., Handbook of Condition Monitoring, Elsevier, 1996, ISBN 1 85617 2341

[2] Iwanitz F. and Lange J., OPC, Fundamentals, Implementation and Application, Hüthig , 2006

[3] Albrecht, R.; Mertins, K.; Sauer, O.: Keine Bindung an den Kundenauftrag. Automobil-Produktion, August 1996, pp. 86-88.

[4] Sauer, O.: Agent technology used for monitoring of automotive production. In: Pham, D.T.; Eldkhri, E.E.; Soroka, A.J.: Intelligent Production Machines and Systems. 1st I*PROMS Virtual International Conference. Oxford: Elsevier 2005, S. 147-153.

[5] FIPA: Foundation for Intelligent Physical Agents
[6] JADE: Java Agent DEvelopment framework
[7] Protégé is a free, open source ontology editor

[5] Bussmann, S.; Schild, K.: Self-organizing manufacturing control: an industrial application of agent technology. Proceedings of the 4th International Conference on Multi-Agent Systems, 2000.

[6] Sundermeyer, K.; Bussmann, S.: Einführung der Agententechnologie in einem produzierenden Unternehmen – ein Erfahrungsbericht. Wirtschaftsinformatik, Vol. 43, No. 2, pp. 135-142, April 2001.

[7] Colombo, A.W.; Neubert, R.; Schoop, R.: A Solution to Holonic Control Systems. Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA Oct. 18, 2001.

[8] Hertzberg, J. et al.: Agententechnologie zur Integration heterogener Systeme am Beispiel eines Prototypen für ein Kommissioniersystem. Proceedings zur SPS/IPC/DRIVES 2003, 25.-27. November 2003.

[9] Urbano, P.G.; Wagner, Th.; Göhrner, P.: Softwareagenten – Einführung und Überblick über eine alternative Art der Softwareentwicklung. Teil 2: Agentensysteme in der Automatisierungstechnik: Modellierung eines Anwendungsbeispiels. Automatisierungstechnische Praxis atp 45 (2003) Heft 11, pp. 57-65.

[10] Bruccoleri, M., Lo Nigro, G., Perrone, G., Renna, P., Noto La Diega, S.: Production planning in Reconfigurable Enterprises and Reconfigurable Production Systems. Annals of the CIRP, Vol. 54/1/2005, pp. 433-436.

[11] FIPA-Standard: www.fipa.org

[12] Bellifemine F., Caire G., Poggi A., Rimassa G.: ***JADE - A White Paper***. http://jade.tilab.com/, version from September 2003.

[13] Stanford University: Protégé-2000 Users Guide. http://protege.stanford.edu/doc/users.html