Agent-based Services using WCF Technology and RFID for Autonomous Control in Continuous Flow Production

Bernd Gastermann and Markus Stopper, *Member, IAENG*

***Abstract*—Manufacturing nowadays is conducted in an environment with prevalently changing conditions. This requires appropriate strategies and systems in order to remain responsive and competitive. Referring to continuous flow production, it is important to focus on cost effectiveness without affecting product quality. Thus, next generation manufacturing systems will need to be able to deal with these requirements. One promising approach to achieve this goal is to increase autonomy of manufacturing systems through software agents. Combined with radio-frequency identification (RFID), agent technology forms a powerful tool to enhance both existing and prospective manufacturing systems.**

This paper proposes an outline of an autonomous control concept based on software agents for distributed manufacturing systems in continuous flow production. Agents and RFID are the groundwork of the concept presented here. The paper provides an overview about both technologies and discusses possibilities on how to combine software agents with RFID for beneficial use in manufacturing. Building on that knowledge, a concept outline with focus on continuous flow production is presented, which consists of distributed agent-based services implemented using Windows Communication Foundation (WCF) and WS-Discovery.

***Index Terms*—agent-based services, continuous flow production, RFID technology windows communication foundation**

I. Introduction

With globally increasing competition, frequently shifting markets and continuous emerging of new technologies, it is vital for prospective manufacturing systems to become more flexible in order to be able to cope with shifting conditions. Among others, such manufacturing systems are expected to seamlessly integrate into heterogeneous software and hardware, to allow extensibility in order to accommodate new sub-systems, to automatically adapt and reconfigure according to the product being worked on, to cooperate with various departments within an enterprise and to quickly react to unexpected faults or changes in order to minimize possible impacts on the working environment.

Agent technology provides a way to satisfy these requirements because it enables manufacturing systems to become both distributed and intelligent. Recently, agent technology has been considered as an important approach for developing industrial distributed systems. It offers a new and more appropriate route to the development of complex computational systems in open and dynamic environments. Particularly in combination with radio-frequency identification (RFID), agents have been recognized as a promising paradigm for next generation manufacturing systems.

Manuscript received December 07, 2010. This work was supported in part by the Automation Research & Development Department of MKW® Austria. Special thanks to the major project promoters Hans and Johannes Danner.

Markus Stopper, Member IAENG, IEEE & DAAAM International, was formerly with the Department of Production Engineering, Vienna University of Technology, Vienna, Austria. He is Honorary Professor at Far-Eastern National Technical University, Vladivostok, Russia and now with the Industrial Automation Research & Development Department of MKW® Slovakia, Prešov, Slovakia (e-mail: markus.stopper@ieee.org).

Bernd Gastermann, Member of DAAAM International, was formerly with the University of Applied Sciences FHSTG Burgenland, Information and Communication Solutions, Eisenstadt, Austria. He is now with the Industrial Automation IT Research & Development Department of MKW® Austria, Weibern, Austria (e-mail: bernd.gastermann@mkw.at).

II. Agents and Multi-Agent Systems

The term "agent" is an abstraction, which generally describes a software entity. It is capable of accomplishing certain tasks in an autonomous way on behalf of its owner or user. Russell and Norvig [12] stated that an agent could be anything that "perceives its environment through sensors and acts upon that environment through effectors" (see Fig. 1). Although the term "agent" is mostly used in conjunction with software, it does not always necessarily refer to a software agent; it may also denote a machine or a person [5]. In context of this paper, though, the term "agent" solely refers to software agents.

Fig. 1. Basic interaction schema between an agent and its environment.

Typical software agents are small, lightweight computer programs that achieve tasks autonomously in an open and dynamic environment. Such tasks are either conducted by

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

agents equipped with the appropriate individual capabilities or by efficient interaction among agents of different types that have complementary capabilities. Agents address autonomy and complexity as they are adaptive to changes, incorporate a certain level of intelligence and are distributed by nature. Interaction between agents could even be regarded as some kind of social activity [7].

According to Franklin and Graesser [5], typical agents differ from ordinary computer programs by characteristics like the ability to respond to the environment, autonomy, goal-orientation and persistence. Persistence in this context means that operation of an agent usually does not stop once its goal is achieved. Programs or subroutines terminate at certain points but agents do not cease to exist when they are done. This implies that agents are capable of continuing their operation and determine their next goals due to their self-reliance and autonomy [5].

A. General Agent Characteristics

Even though different definitions from various authors exist, agents are generally defined by the following characteristics [3]:

- *Autonomy:* Functionality of an agent is independent from human-interaction or other agents. Some agents have individual internal states and goals, and are also able to make their own decisions. They act in such a manner as to meet their goals on behalf of their users and are thus operating autonomously.
- *Social ability / collaboration:* Agents may communicate with other agents or components using common semantics or some kind of agent-communication language (ACL). They may even collaborate with one another on a common task. Although not required, cooperation is especially important for mutual benefit when multiple agents operate in a shared environment (i.e. in a multi-agent system).
- *Reactivity / intelligence:* Agents perceive their environment and respond appropriately to changes or events that occur in it (see Fig. 1). Certain agent types (cognitive agents, for instance) use existing, predefined knowledge to achieve their goals. Some are also able to learn from the experience they gain while they react to or interact with their external environment [2].
- *Pro-activeness:* Most agents do not simply act in response to their environment but are also able to exhibit opportunistic, goal-directed behaviour by taking the initiative when appropriate. Thus, they are capable of generating and pursuing their own goals.

However, the provided list of characteristics does not claim completeness. Different authors come up with their own interpretation as there is no general agreement on a definitive list of agent attributes. Characteristics are also dependent on the agent's field of application, which means that some attributes may not apply to certain types of agents. For instance, agents in a multi-agent system do not necessarily have to be benevolent to one another. Cooperation serves no purpose when in competition with other agents [2].

Besides, agents are classified into three types of agent architectures, which describe the mode of operation with respect to agent behaviour:

- *Reactive agents:* Reactive agents are not provided with models of their environment. They do not have reasoning capacity that involves more than just environment perception. Behaviour of reactive agents is determined by a simple stimulus-response pattern while reacting to messages from the environment or other agents. This means, that they respond to the present state of the environment in which they are embedded by simple input-output rules. The advantage of reactive agents is that they are much easier to implement than cognitive agents but their capabilities are more limited [1]. However, while being more efficient than cognitive agents to solve generic and simple tasks, reactive agents are generally not as versatile as cognitive agents [4].
- *Cognitive agents:* Contrary to reactive agents, cognitive agents (which are sometimes also called deliberative agents [2]) carry an explicit representation of their environment and are thus more powerful than reactive agents. Cognitive agents are based on knowledge (or a set of beliefs), desires and intentions (BDI model) instead of behaviour. In particular, desires represent the state of environment the agent prefers whereas intentions represent the state of environment the agent tries to achieve. Activities are based on action plans which are dynamic and constantly changing according to the agent's memory of past actions, goals and future plans. A cognitive agent uses sensors to perceive actions in the environment, assesses the current state and eventually predicts a future state of the environment based on his beliefs, desires and intentions. As it has different possibilities to solve a problem, the agent makes a decision based on this assessment in order to act in the environment using its effectors (see Fig. 1) [4]. As focus lies on symbolic reasoning, negotiation and planning, cognitive agents are able to accordingly anticipate future events and apply logical inference in order to satisfy their goals.
- *Hybrid agents:* These are a mixture of both cognitive and reactive components. Hybrid agents attempt to balance both reactiveness and deliberativeness.

Furthermore, agents can be classified by mobility, which defines whether they are able to move around some network or not. This yields the classes of static and mobile agents [2].

B. Multi-Agent Systems

When adopting agent technology, a single agent is often not sufficient to cope with certain complex tasks (if they only have a partial model of their environment, for example). Thus, multiple agents are usually used in the same common environment to form a network of communicating, interacting and collaborating agents. Such an environment or system, in which several agents communicate and interact with one another, is called multi-agent system (MAS).

Within a multi-agent system it is especially important for agents to incorporate mechanisms which allow them to

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

synchronize and coordinate their activities at runtime. Such mechanisms are coordination models that provide both media and rules for managing interactions. Coordination requires regulated flow of information, that is: communication using some common language [7].

III. Radio Frequency Identification

Radio-frequency identification (RFID) is a generic term for technologies that use radio waves for the identification of animals, humans or objects. The RFID system comprises a reader and a transponder component (also referred to as tag) which is associated with physical objects (see Fig. 2). Identification of objects is performed on the basis of a unique identifier, which is stored on the tag [6]. The tag itself consists of at least an integrated circuit and an antenna. The RFID systems basic operating principle is energy and data transfer using propagation of radio signals. In general, RFID is similar to barcode systems. The main difference is that barcodes use optical identification and therefore must be in direct line of sight with the reader whereas RFID tags must not. Additionally, RFID tags are more resistant to various environmental influences like rain, snow, dirt, oil, paint, etc. It is possible to read up to hundreds of RFID tags simultaneously – even through objects. This does, however, depend on the type of RFID tag being used [6].

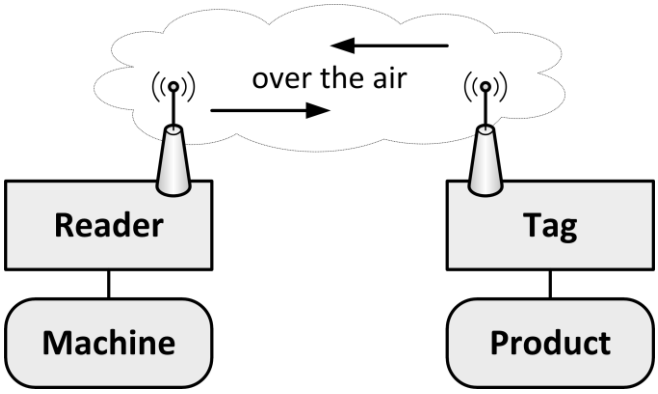

Fig. 2. Basic working principle of the RFID system. The figure shows reader and tag exchanging data over a common air interface.

RFID tags are typically characterized by features like communication frequency, power supply and the ability to store and/or process information [10]. Having various configurations, tags can either be read-only or also be able to store custom data. Available storage capacities range from a few bits up to several megabytes. Basically, there are two common types of RFID tags [6]:

- *Passive:* Passive RFID tags are designed to be small and low cost. A passive tag only contains an antenna and circuitry that stores data, but does not include an internal power source. Energy required for sending data back to the reader is detracted from the electro-magnetic energy field generated by the reader while trying to read the tag information. This technique is also called "backscattering". However, this approach makes it suitable for short-range communication and small data sizes only.
- *Active:* In contrast, active RFID tags use their own internal battery to send data. Their battery allows them to transmit data at a greater range (up to several hundred meters). Active tags are also able to continuously or periodically broadcast signals and data, regardless of whether the tag is in the field of a reader or not. In this case the tag acts as a beacon. Depending on the configuration of the tag, it may also have a processor, read-only memory (ROM), which holds the firmware, or memory to store user-defined data. Admittedly, such additional components make active tags also the most expensive ones.

In addition to passive and active RFID tags, there are also two hybrid forms: *semi-passive* and *semi-active* tags. Just like active tags, both types include an internal battery. However, semi-passive tags do not use their internal battery to send data back to the reader. Although it is still using the "backscattering" mechanism, the purpose of the battery is to keep the tag powered even when it is not being read. This allows the tag to continuously monitor its environment and process data, for example. Energy required for data transmission is still induced by the reader as it attempts to read the tag. On the other hand, semi-active tags use the battery for data transmission, but they do not remain active all the time. Instead, they have to be activated by a low frequency signal coming from the reader. Then, they are switching back to a sleeping-mode when not used. Compared to active tags, lower costs and longer battery life are motives for using semi-passive or semi-active tags [6].

In general, a RFID tag uses one of the following communication frequencies [8]: low-frequency (LF), high-frequency (HF), ultra-high-frequency (UHF) or microwave frequency. Tags that use low-frequency communication typically operate at frequencies of 125 or 134.2 kHz, which results in communication ranges below half a meter. High-frequency tags use 13.56 MHz and reach communication ranges of up to one meter. Ultra-high-frequency tags operate in a frequency range of 860 to 960 MHz, resulting in effective communication distances from one to ten meters. LF and HF tags are usually passive, which means that they do not need a battery for communication. Tags that use microwave band operate at 2.45 or 5.8 GHz. Their communication range varies from several meters up to a hundred meters. Due to the high frequency, read speed of microwave frequency tags is generally better than with any lower frequency tag. Higher frequency tags like UHF and microwave band tags are typically active as they require more energy for sending. Because of international restrictions, not all of these frequency bands are available in all countries.

IV. Agents and RFID in Manufacturing

Agents become increasingly important in manufacturing because they help to implement important characteristics such as autonomy, responsiveness, redundancy, distribution and openness. Many tasks related to manufacturing could be conducted by agents. Hence, combining agents with RFID

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

tags on products is a natural progression which increases flexibility and scalability in production. Doing so allows each agent to autonomously react to various influences and configure production line parameters according to the particular product.

The multi-agent system architecture of PABADIS'PROMISE – a research project of the European Union – connects shopfloor level, Manufacturing Execution System (MES) and Enterprise Resource Planning (ERP) system using agents. ERP is the level responsible for taking orders, which are then processed and executed at MES and shopfloor level. In PABADIS'PROMISE, a multi-agent system lies at the core of MES which generally consists of order agents and resource agents. Machines and devices of the MES are represented by resource agents, which in turn provide services to product agents. Order agents are the intelligence of the system. They execute orders and decide on how to reach their goals best [6]. This project also makes use of mobile software agents which are located on RFID tags attached to products [11].

Depending on the capabilities of the RFID tag being used, there are the following four ways of implementing agents on RFID tags [9]:

A. *Product Identification Tag*

A simple Product Identification Tag (PIT) is the most cost effective approach as it uses a passive RFID tag that only provides a unique identification (ID) number. Using this ID, the associated product agent that handles further operations needs to be loaded from the network at each step of the production line. There are a few drawbacks, though: With passive tags, communication range is limited (about one meter) and storage for custom data is usually not available. Therefore, agent code and product data are physically separated from the product itself and have to be loaded from the network each time the product is processed [10].

B. *Product Data Tag*

The Product Data Tag (PDT) is a slight enhancement to the PIT. It does not only carry an ID but also stores order and other product specific data. Depending on the amount of storage capacity required, passive or active RFID tags could be used. It must be noted that active tags and increasing memory will make this approach more expensive than the first one, however. The agent itself, on the other hand, still needs to be loaded from the network and executed in an appropriate runtime-environment which could be located at each production step, for instance. Nonetheless, this approach saves network traffic and is more flexible as it allows carrying all necessary production data directly on the product itself. This is especially useful in a production process which is distributed across several facilities or which requires many human interaction steps [10]. However, this fact could also become a major drawback, especially when using passive tags: Permanent access to production data is not guaranteed as the data only becomes available when in proximity of a reader.

C. *Product and Agent Tag*

Another approach called Product and Agent Tag (PAT) contains ID, product data and agent code. Although the tag stores the agent code, it is not able to execute it right away. This saves network traffic and guarantees a strict, physical coupling with the product but also means that the agent will remain inactive most of the time and thus cannot schedule, process or monitor production tasks. Agent code will only execute once loaded by a machine. This makes this approach unsuitable for production processes with long transportation times (e.g. on a conveyor line). However, it allows self-contained execution as no additional data is required to be loaded from the network.

D. *Product and Agent Host Tag*

A Product and Agent Host Tag (PAHT) requires more resources (in terms of processor and memory capacities) than any of the previous approaches, which also make it the most expensive one. Its huge advantage is that it allows direct execution of agent code and thus represents the most ideal implementation of the agent concept as the agent always runs regardless of its current location. Agent and product data are tightly coupled with the product, which gives maximum flexibility in production. The agent is able to communicate with other agents and to constantly monitor production progress regardless of its current location, enabling it to react to certain conditions and events. However, this approach can only be achieved by active RFID tags also capable of processing data. Due to the high costs involved with these tags this solution is not suitable for continuous flow production.

V. DISCUSSION

In continuous flow production, cost effectiveness is essential as large amounts of products have to be considered. Thus, given the four previously introduced possibilities of how to implement agent technology on RFID tags, only PIT seems appropriate for continuous flow production from an economical perspective, although PDT and PAT could also be used under certain circumstances. Nevertheless, the concept provided here in this paper is restricted to PIT and will only take this approach into account. Using PIT, however, has the implication that tagged products are required to be in the right position and not too far away from the reader in order to be scanned successfully. Also, the product will only be identifiable using the RFID tag's unique identifier whereas additional product information and agent code have to be acquired from an external source. Adopting PIT implies that agent and product information are physically separated from each other.

The concept presented in this paper is specifically tailored for use in continuous flow production with independent production lines. In general, it describes a multi-agent system distributed across MES and shopfloor level. ERP is not discussed here, but it still represents an essential part as all orders are eventually coming from the ERP system. The main purpose of the concept is to provide flexibility in

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

production control and to allow autonomous control of manufacturing systems by reducing overall complexity and decentralizing intelligence by spreading it into dispersed agents. As agents are equipped with mechanisms for scheduling and resource allocation, they are able to perform tasks rather autonomous and independent from any centralized planning component.

The presented concept uses (intranet) web service technology like Windows Communication Foundation (WCF) in order to implement agents and other basic services required for operation. The decision to use WCF has primarily been made because it already offers established and standardized ways of communication and allows easy integration into existing manufacturing infrastructures, which may already use service-oriented architectures and are thus also having most of the required infrastructure. Using a service-oriented architecture enables the agents to be executed in separate runtime environments, which avoids downloading agent code to the machine. Instead, agents are able to communicate using remote web service calls, regardless of the actual location in the system.

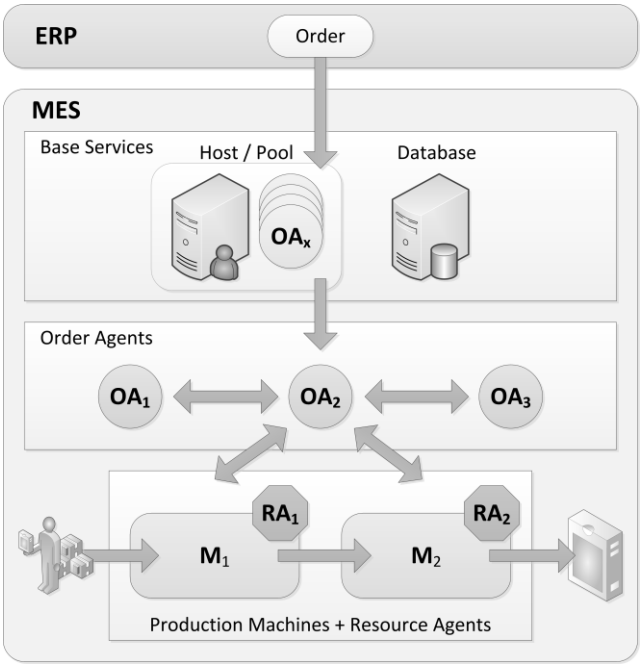

Fig. 3. Overview of the core components of the system. Shows how incoming orders from ERP are processed in a top-down approach.

A. Concept Design and Core Components

The conceptual system generally consists of two basic web services and two types of agents (see Fig. 3). The function of the services is to provide the ERP an interface to the multi-agent system and to provide agents with data. The agents themselves incorporate the intelligence of the system. Both agent types are implemented as reactive agents. Cooperation between them is essential in this concept. As single agents only have limited knowledge and resources, agents must collaborate with each other in order to achieve their goals. The advantage, however, is that scheduling is local to one agent, so changes can easily be done without influencing other agents. Because there is no coupling between agents, they can easily be replaced in case of a failure. Another benefit is load-balancing, which is done automatically as agents request specific production abilities instead of addressing a certain resource directly. Due to the loose coupling, a look-up mechanism is required which allows the agents to find one another. The multi-agent system presented in this paper uses WS-Discovery – a multicast protocol using SOAP-over-UDP to locate services on a network. WS-Discovery replaces the centralized "Ability Broker" described in PABADIS'PROMISE with a more distributed approach. Retrieval of agent addresses and abilities is done solely via WS-Discovery although a centralized broker service could also be used. However, the distributed approach has been given preference because the concept in this paper aims to use as few centralized services and external dependencies as possible. The only disadvantage with this approach would be that network traffic could increase in huge networks [9].

The following section introduces each component of the system (Fig. 3) in more detail:

Agent Host and Pool Service

The Agent Host and Pool (AHAP) service primarily serves two purposes: It acts as a host (runtime environment) for all currently active order agents, and it acts as a pool that harbours inactive order agents until they become active. The service is part of the interface between the ERP and the multi-agent system. All incoming orders from ERP are processed here. For each new order, the service is responsible for the creation and initialization of order agents. It is furthermore responsible for the supervisory control of their life cycles. During runtime the service continuously collects information about the production process and progress of all hosted order agents. This information is then stored in a database using the DB service, for example.

Database Service

The database (DB) service provides database access for all components of the system. Its purpose is to encapsulate one or more databases and provide transparent data access for other components. The service yields material-, order- and product-related information. Using this information, the agents gain knowledge on how to produce a product. This includes, among others, optimal machine configuration parameters and required manufacturing processes. Additionally, the service is also used to store various log data, which is required for monitoring and tracing purposes.

Order Agent

An order agent (OA) represents an active, currently running order from the ERP system and implements crucial MES functions like scheduling and resource allocation. These order agents are the core component of the system and incorporate the business logic required to autonomously organize and execute the order they represent. They are responsible for the execution control of a production order. An OA makes decisions only for the products in scope of its current order. Instead of creating an agent per product, there is only a single agent per order in this concept. The reason for this decision was to decrease the overall amount of agents concurrently running in the system. Interaction

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

between order agents allows for efficient utilization (load balancing) of each machine. It also enables the system to quickly react to certain changes like maintenance or failures in order to sustain an optimal production process.

Resource Agent

The resource agent (RA) is the counterpart of an order agent. These agents are tightly coupled with production devices like robots or conveyer lines. Each resource agent is responsible for the management of a resource. The agent directly resides on the resource it represents to the MAS and runs on an attached embedded system. Resource agents are the link between the multi-agent system and the field device controlled by it. At start-up the agent initializes the resource and registers itself with the other agents of the system using WS-Discovery. Registration includes data about its abilities as well as its availability. More precisely, abilities tell other agents which products can be processed and how. The agent is also responsible to process requests from order agents and to request activation of an order agent from the AHAP service. Furthermore, the RA needs to schedule and allocate its underlying resource and supervise possible exceptions. In case of exceptions, it has to inform relevant order agents.

Production Machine

The production machine is the field device that actually processes products or raw materials. Each has certain abilities which are relevant for the production process. For the system, the machine is represented by a RA that controls machine parameters and provides certain services, which are based on the machine's own abilities, to other agents.

B. Workflow Example

Given the design and the components of the multi-agent system, a basic workflow scenario will be outlined in the following section:

Every order from ERP consists of a certain number of products, each identifiable by a unique ID on a passive RFID tag attached to the product (PIT). In turn, each of these articles is being made from raw materials, which are also tagged using PIT. Whenever a new order is received, it is recorded in the ERP system, which then instructs the AHAP service to create a new order agent specifically for this new order. To do so, the service has to acquire information about the order, its products, raw materials and production steps required to produce them. This data is obtained from the database service. Having gathered all the information, the order agent is instantiated. However, it initially remains in an inactive, sleeping state. This is the case because it is unlikely that production of a new order is started immediately after it has been placed. If required, starting the order agent right after its creation is possible nonetheless. Inactive agents wait on the agent pool of the AHAP service until activated.

Resource agents are able to activate a specific order agent. Doing so becomes necessary if a resource agent reads the tag of raw material that belongs to a yet inactive order agent. In this case, the resource agent issues a request to the AHAP service to start the appropriate agent. From this moment onwards, this agent starts operation and handles production of all products belonging to that specific order. It calculates schedules and allocates resources in coordination with other running agents, instructs resource agents to set specific machine parameters and organizes transportation between production steps.

VI. Conclusion & Further Research

The solution proposed in this paper should be considered as an architectural and technological concept. Note that prototyping still has to be done in order to verify the design of the concept presented in this paper. Although it tries to cover as many aspects as possible, it is far from being complete. The scope considered in this paper is only with continuous flow production and multiple independent production lines. Other application areas are not covered yet. Additionally, language required for communication between agents and fault tolerance with regard to RFID is also not covered in this paper. Further research still has to be done in these areas as well as in the concrete implementation details for agents and services. Besides, the effects of WS-Discovery on scalability remain to be investigated.

References

[1] A. D. Pace, F. Trilnik, M. Campo and A. Clausse. (2001). *Object-Oriented Simulation of Fluids using Software Agents* [Online]. Available: http://tiny.cc/ojkwb/

[2] H. S. Nwana. (1996, September). Software Agents: An Overview. *The Knowledge Engineering Review* [Online]. 11(3). pp. 1-40. Available: http://agents.umbc.edu/introduction/ao/

[3] M Wooldridge and N. R. Jennings. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review* [Online]. 10(2). pp. 115-152. Available: http://tiny.cc/p1akr/

[4] M. A. S. N. Nunes, L. L. Dihl, L. Fraga, C. R. Woszezenki, L. Oliveira, D. J. Francisco, G. J. C. Machado, C. R. D. Nogueira and M. G. C. Notargiacomo. (2002). Multi-agent Systems Applied to Intelligent Tutoring Systems. *Proceedings of the International Conference on Artificial Intelligence; IC-AI '02* [Online]. pp. 361-366. Available: http://tiny.cc/m0h1q/

[5] S. Franklin and A. Graesser. (1996) Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents. *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages; ECAI '96* [Online]. 1193. pp. 21-35. Available: http://tiny.cc/vy6v5/

[6] PABADIS'PROMISE Consortium. (2006, June 15). *Definition of Overall PABADIS'PROMISE Control System Architecture* [Online]. Available: http://tiny.cc/6eegy/

[7] L. Monostori, J. Váncza and S. R. T. Kumara. (2006). Agent-Based Systems for Manufacturing. *CIRP Annals - Manufacturing Technology* [Online]. 55(2). pp. 697-720. Available: http://tiny.cc/ljvaf/

[8] L. Ivantysynova. (2008, October 24). *RFID in Manufacturing: Mapping the Shop Floor to IT-Enabled Business Processes* [Online]. Available: http://tiny.cc/w4rlk/

[9] PABADIS'PROMISE Consortium. (2006, September 13). *Concept of Overall PABADIS'PROMISE Architecture Components* [Online]. Available: http://tiny.cc/30jjg/

[10] A. Treytl, A. Bratukhin and G. Pratl. (2006). Product Identification in Distributed Production Processes using RFID and Agent-Based Control. *Proceedings of Product Identification in Distributed Production Processes using RFID and Agent-Based Control 2006* [Online], pp. 415-423. Available: http://tiny.cc/556ul/

[11] PABADIS'PROMISE Consortium. (2005). *PABADIS'PROMISE Project Description* [Online]. Available: http://tiny.cc/o2vmo/

[12] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 2nd Edition*. Upper Saddle River, New Jersey: Prentice Hall, 2003, ch. 2.

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)