# Embedded Autonomous Agents in Products Supporting Repair and Recycling

Leo van Moergestel, Erik Puik and Daniël Telgen
HU Utrecht University of Applied Sciences
Utrecht, the Netherlands
Email: leo.vanmoergestel@hu.nl
erik.puik@hu.nl
daniel.telgen@hu.nl

John-Jules Meyer
Utrecht University
Utrecht, the Netherlands
Email: jj@uu.nl

***Abstract*—This paper describes a concept where products are equipped with agents that will assist in recycling and repairing the product. These so-called product agents represent the product in cyberspace and are capable to negotiate with other products in case of recycling or repair. Some product agents of broken products will offer spare parts, other agents will look for spare parts to repair a broken product. On the average this will enlarge the lifetime of a product and in some cases prevent wasting resources. Apart from reuse of spare parts these agents will also help to locate rare elements in a device, so these elements can be recycled more easily.**

***Keywords*—*product agent; agent based recycling; green IT***

## I. Introduction

Agent technology for agile manufacturing was the starting point of this research. In this research about agile manufacturing the concept of a product agent was introduced. Every product to be made starts as a software entity or agent that is programmed to meet its goal: the production of a single product. To be able to reach its goal this agent knows what should be done to create the product. This product agent guides the product along the production cells to be used for manufacturing and it will collect all kinds of important manufacturing data during the production process. When the product is finished, this agent has all the manufacturing details and is still available for further use containing valuable data about the product. The next step in this approach is to investigate and study the roles of this product agent in the other phases of the life cycle of the product.

In this paper the focus will be on the role of the product agents during repair and recycling. After the description of what an agent is and what role it can play in the life cycle of a product, a simulation of the part exchange model for a certain situation is given. Next an autonomous distributed agent-based system will be described that has been implemented to exchange spare parts using a marketplace in cyberspace. Finally the role of the product agent in recycling will be discussed.

## II. Agents

There are many definitions of what an agent is. We use here a commonly accepted definition by Wooldridge and Jennings [28] *An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives.* In figure 1 we depict an agent in its environment. An agent is sensing the environment and can perform actions on the environment. As stated in the definition, the actions the agent performs depend on the design objectives.

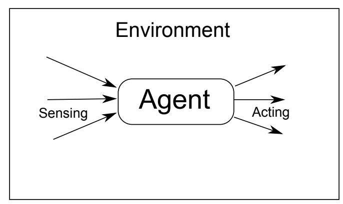


Fig. 1. An agent in its environment

In figure 1 the agent is a black box, so now we must take a look at the internal software structure of an agent. To do this we should first discuss the possible implementations of agents, but this is too broad a field to handle here. We will concentrate on some aspects that are important for our final software architecture. Literature and papers about agents introduce among others, two types of agents:

1) reactive agents
2) reasoning agents

A reactive agent senses the environment acts according to the information it gets from this sensing. There is no internal state involved. A reasoning agent also senses its environment but does have an internal state. Depending on the sensing input and the internal state it will search for an action to perform, one could say it will reason for the action to perform. The sensing input will also change the internal state. A special type of reasoning agent is the so called belief-desire-intention-agent or BDI-agent. This type of agent has its backgrounds in the philosophies of Dennett and Bratman [10][3]. An internal schematic of a BDI-agent can be seen in figure 2 [27].

The beliefs, desires and intentions could be viewed as the mental states of a BDI-agent.

- from the inputs of its sensors the agent builds a set of *beliefs*. Beliefs characterize what an agent imagines its environment state to be;
- *desires* (or goals) describe the agent's preferences;

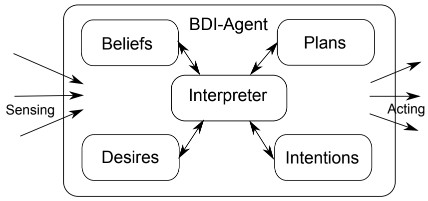


Fig. 2. BDI-agent

- *intentions* characterize the desires the agent has selected to work on.

An agent is equipped with a set of *plans*. These plans have three components:

1) the postcondition of the plan;
2) the precondition of the plan;
3) the course of action to carry out.

An agent will deliberately choose a plan to achieve its goals.

### A. Multi-Agent Systems

A multi-agent system (MAS) consists of two or more interacting autonomous agents. Such a system is designed to achieve some global goal. The agents in a multi-agent system should cooperate, coordinate and negotiate to achieve their objectives. When we consider the use of a multi-agent system the notion of a role is an important aspect. A role is the abstract representation of a policy, service or function [13]. Within a role, certain abstract concepts should be specified [11] such as:

- *permission*: what are the constraints the agent is tied to;
- *responsibility*: i.e. the responsibility an agent has in achieving the global goal;
- *interaction*: agents interact with each other and the environment

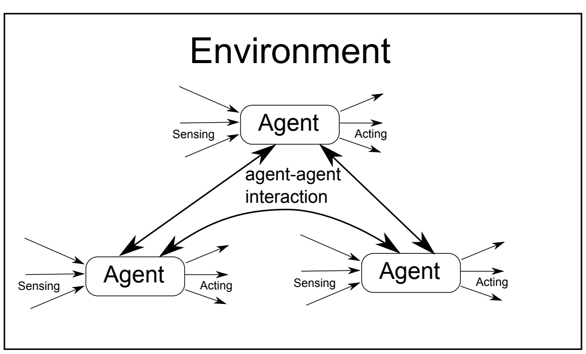


Fig. 3. Multi-agent system

The properties of a single agent properties are perfect for its role as a software entity that guides and monitors a product. The agent is autonomous, reactive and can be pro-active. In a MAS these agents can be cooperative and should communicate. This situation will play an important role in case of exchange of parts during repair and recycling as will be explained in this paper. In that case a software infrastructure is needed where agents can play a specific role. These agents play their role and interact with other agents. Generally in a MAS there are interactions at two levels: first the actions of the agent within the environment and second the interaction between agents.

## III. ROLE OF AGENTS IN THE LIFE CYCLE OF A PRODUCT

In figure 4 the life cycle of an arbitrary product is shown. After the design, the product is manufactured in the production phase, next the product is distributed. A very important phase is the use of the product and finally the product should be recycled. In all of these phases, the product agent can play a role that will be globally described in the next sections


Fig. 4. Life cycle of a product.

### A. Design and Manufacturing

In our view the design of a product will be greatly influenced by the individual end user requirements. This means that cost-effective small scale manufacturing will become more and more important. In [26] and [20] a manufacturing system based on a grid of cheap and versatile production units called equiplets is described that is capable of agile multiparallel production. In this model every single product is guided through the production environment by the already introduced product agent. This agent is responsible for the manufacturing of the product as well as for collecting relevant production information of this product. This is normally a function of the so-called Manufacturing Execution System (MES) [18]. The result is that every product has its own production journal in contrast to one journal for a whole batch of products. In figure 5 the agent based manufacturing is depicted. In this figure the product agent is hopping from equiplet to equiplet to guide the product along the production machines or equiplets and monitor success or failure of the production steps [21]. To

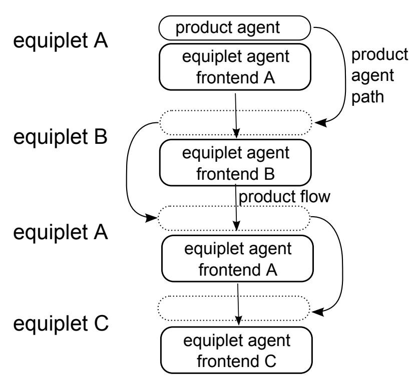


Fig. 5. Product agent and equiplet agents during production.

make a smooth transition from design to production possible and a short time-to-market, the product agent is designed as a co-design for the product. Though this is all based on our own special production environment, we expect this approach to be useful in other production environments as well. The roles of the product agent are: production scheduling (in a MAS with other product agents), guide the product along the production equipment, inform the production equipment (equiplets) about the manufacturing steps, collect manufacturing data, collect relevant product data.

The concept of using agents for production is not new. Among others a multiagent-based production system has also been developed by Jennings and Bussmann [7]. This system focuses on reliability and minimizing downtime in a production line. This approach is used in the production of cylinder heads in car manufacturing. The roles of the agents in this production system differ from our approach. This has to do with the fact that Jennings and Bussmann use agent technology in a standard pipeline-based production system and the main purpose was to minimise the downtime of this production system.

### *B. Distribution*

Product agents can, in their role of negotiator, communicate with logistic systems to reach their final destination. Logistic applications based on multi agents systems already exist [6]. Information of product handling and external conditions, like temperature, shocks etcetera can be measured by cheap wireless sensors and collected by the guidance agent during the transport or after arrival at the destination. The handling and external conditions during transport can be important during product use, especially for product quality, maintenance and repair.

### *C. Use*

The role of the product agent during the use of the product could focus on several topics. The first question one should ask is: who will benefit from these agents, i.e. who are the stakeholders. In a win win situation both the end user as well as the manufacturer could benefit from the information. If a product is a potential hazard (in case of misuse) for the environment, the environment could also be a winner if the agent is capable to minimize the effects of misuse or even prevent it.

*1) Collecting information:* A product agent can log information about the use of the product as well as the use of the subsystems of the product. Testing the health of the product and its subsystems can also be done by the agent. These actions should be transparent for the end user. If a product needs resources like fuel or electric power, the agent can advise about this. An agent can advice a product to wait for operation until the cost of electric power is low i.e. during the night.

*2) Maintenance and repair:* Based on the logging information about the product use and the use of the subsystems, an agent can suggest maintenance and repair or replacement of parts. Repairing a product is easier if information about its construction is available. Also the use of a product or the information about transport circumstances during distribution can give a clue for repair. An agent can also identify a broken or malfunctioning part or subsystem. This could be achieved by continuous monitoring, monitoring at certain intervals or a power-on self test (POST).

An important aspect of complex modern products are updates or callbacks in case of a lately discovered manufacturing problem or flaw. In the worst situation, a product should be revised at a service center or the manufacturing site. Information about updates or callbacks can be send to the product agent that can alert the end user in case it discovers that it fits the callback or update criteria. This is a better solution for a callback than globally advertising the problem and alert all users of a certain product when only a subgroup is involved.

*3) Miscellaneous:* Use of product agents could result in transparency of the status of a product after maintenance by a third party. The agent can report to the end user what happened during repair so there is a possibility to check claimed repairs. Of course the agent should be isolated from the system during repair to prevent tampering with it. Recovery, tracking and tracing in case of theft or loss are also possible by using this technique.

### *D. Recycling*

Complex products will have a lot of working subsystems at the moment the end user decides it has come to the end of its life cycle. This is normally the case when a certain part or subsystem is broken. The other remaining parts or subsystems of the product are still functional, because in a lot of complex products the mean times between failure (MTBF) of the subsystems are quite different. The product agent is aware of these subsystems or components and depending on the economical value and the remaining expected lifetime these components can be reused. This could be an important aspect of 'green manufacturing'. An important issue here is that designers should also take in account the phase of destruction or recycling. Disassembly and reuse of subsystems should be a feature of a product for this approach to be successful. This paper shows some results of a simulation where recycling of subsystems is applied.

The product agent can reveal where rare or expensive material is situated in the product so this material can be recovered and recycled. This way the product agent can contribute to the concept of zero waste. *Zero waste is just what it sounds like - producing, consuming, and recycling products without throwing anything away* [16].

The roles of the product agent in this phase are: locating broken parts, negotiate with other agents for part exchange, revealing positions of expensive or rare parts or materials within the product.

## IV. RESOURCE DEPLETION

A problem humanity encounters is the depletion of natural resources. This can be seen by the sometimes enormous increase of price of some of these resources. The price of lead, gold and copper increased by resp. 378 %, 308 % and 269 % from 1999 to 2009 [1]. In some cases this has to do with the use of elements that are hard to find. In other cases the demand for elements has increased because of certain newly developed applications or an increased field of applications for that element. Apart from searches for new places to mine these elements, another - for the environment perhaps better way - to come around this problem is to reuse material. Today cellphones containing these rare elements are considered a new kind of ore. To reuse the elements it would be nice if it could be located within the device. This is where the product agents comes in handy. Table I shows some of these rare elements

TABLE I. DEPLETION OF ELEMENTS

| Name | Symbol | Years available |
|---|---|---|
| Silver | Ag | 29 |
| Indium | In | 13 |
| Antimony | Sb | 30 |
| Hafnium | Hf | 10 |
| Tantalum | Ta | 116 |

and the expected time left before recycling is the only way to get these elements [8]. The expected time left is based on the use as it is. If the use of a certain element increases, the time left will be shorter of course. Product agents will help us hunting for places where rare metals are concentrated enough to be worth recovering. This is because the product agent carries all the information that has been collected during the production phase. To make this concept work, a list of “ingredients” should be part of this information.

## V. PRODUCT TYPES

This approach of having an agent for a product could be used on different kind of products, but one should investigate if the final product has intelligence and hardware to communicate with the agent. Some products have this by nature (computers, cell-phones); for other products (cars, machinery, domestic appliances) it should be a small investment.

### A. Where do these agents reside?

A product agent should stay alive or at least the information the agent has collected and the knowledge the agent has learned should be available under all circumstances. To accomplish this, two solutions are available. The agent can be a mobile agent moving from platform x to platform y as depicted in figure 6a. The other solution requires moving data (beliefs of the agent) from one agent to a newly created agent as shown in figure 6b. In our case both agents should be product agents. The second solution is much easier to implement because of

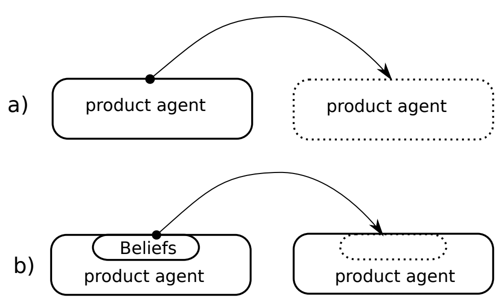


Fig. 6. Mobile agent versus moving data.

the fact that only transport of data is required, while in the case of moving agents, the whole executable should be adapted to the new situation. Another advantage of the second approach is that a product agent can be added in any phase of the life cycle. This is also what has been done for this specific research. A product agent was added to a system in the use phase. The biggest challenge for implementing the approach of a product agent or guidance agent will be in the use phase. This is where the product is under control of the end user. This is a divers environment that is difficult to predict. The environment during the production under control of the manufacturer is not that divers and will be rather predictable. In the latter case an agent-based infrastructure can be implemented for the production system or production line. The same is true for transport and even disassembly of the product. In case of the use phase, the agents should reside in a system that is connected to the product, but should be available at the moment the product itself is broken. This is comparable to the case of the so-called black box in aeroplanes. There are several possibilities, depending on the type of product:

- The agent runs on its own separate hardware that is closely tied to the product;
- The agent runs on the hardware of the product but stores information on a special place on the product itself. This information can be recovered after breakdown;
- The agent runs on the hardware of the product but stores information on a remote system;
- The agent runs on a remote system that has a continuous connection;
- The agent runs remote on a system using a ’connect when necessary’ approach.

The last two options require a stub or entry point for the remote agent to make contact with the product system. The connection with the environment could be established by wired or wireless sensors or sensor networks as well as computer subsystems in the product. Interaction with humans in the environment could be established by a messaging system or human computer interface (HCI).

## VI. RECYCLING OF SUBSYSTEMS

In this section a Monte Carlo simulation is used to show the effect of recycling subsystems of broken products. Also a marketplace model is presented that has been developed to show a possible implementation of agent-based recycling.

### A. Extending the average lifetime

An interesting application for the product agent can be automatic recycling of subsystems during its use. To explain this in more detail, consider a product consisting of 2 equal subsystems. This means that these subsystems have the same average lifetime and are also similar. In this section we focus only on products where two or more subsystems are all of equal type. We assume that the lifetime is a normal distribution according to formula:

$$f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\mathrm{e}^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

Where $\mu$ is the average lifetime, $\sigma$ the standard deviation and $\sigma^2$ the variance. An end user considers the product to be broken if one of these subsystems is broken. Normally one subsystem will be the first to fail leaving another still functioning subsystem in the broken product. To get some insight in this situation, a Monte Carlo simulation was set up. This simulation was based on 1000 products, starting with two equal subsystems. For every product we generated two failure times according to the given normal distribution. The plot of all these failure times (see figure 7) turned out to be a rough approximation of the theoretical Gaussian curve. If we plot the minimum and maximum failure time of the two equal systems per product, this results in two smaller rough approximations of Gaussian curves as depicted in figure 8.

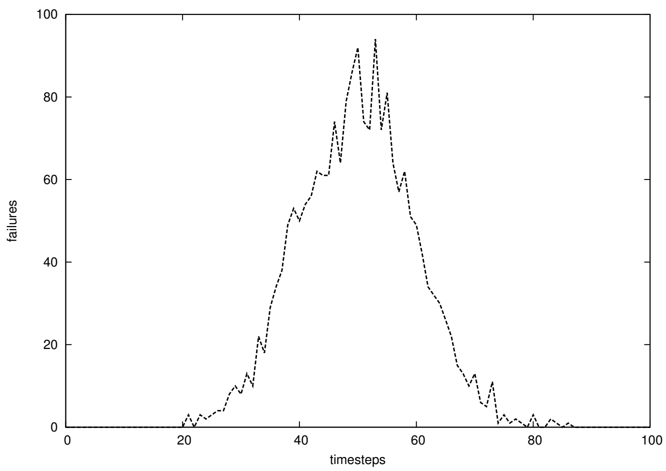


Fig. 7. Distribution of failing subsystems

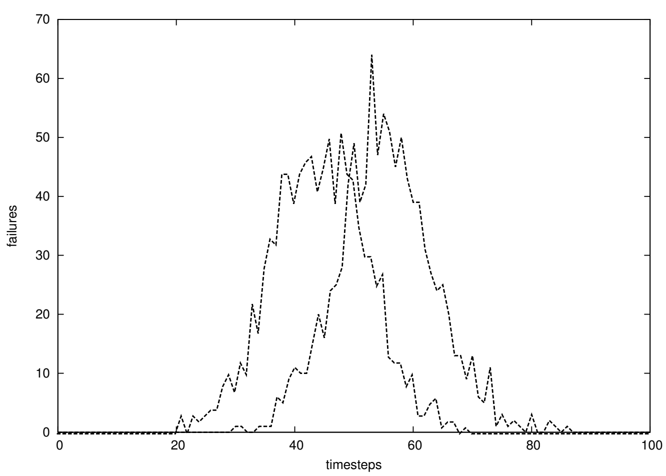


Fig. 8. Distribution of two failing subsystems

If there is no exchange of subsystems among the products, half of the products will be broken on the average time of the Gaussian curve belonging to the first subsystem to break. Taking the average of the left curve in figure 8, this turns out to be at time $t = 432$. If exchange of subsystems is possible, half of the products will still work at $t = 500$. The gain in lifetime will be: $\frac{500-432}{432}$ and could also be expressed as a percentage. The gain for this situation will be bigger if there are more equal subsystems in the product. In figure 9 the gain in average lifetime is plotted as a function of the number $n$ of equal subsystems. A value of 100 for $\sigma$ and 500 for $\mu$ is used. Starting with $N = 1$ there is no gain at all, because on the average 50% of the products will be broken at time $\mu = 500$.

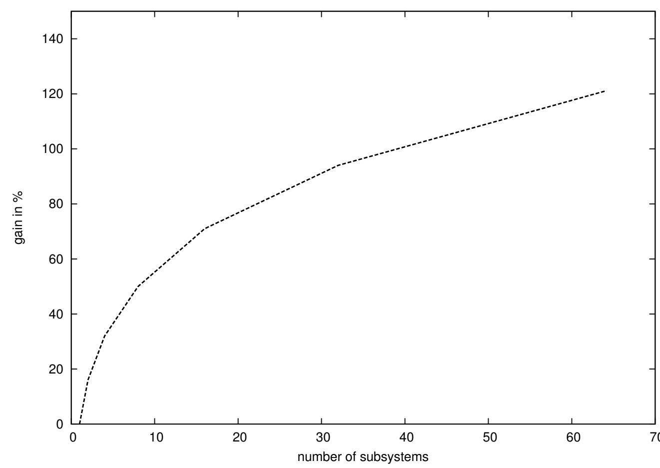


Fig. 9. Gain as a function of the number of subsystems

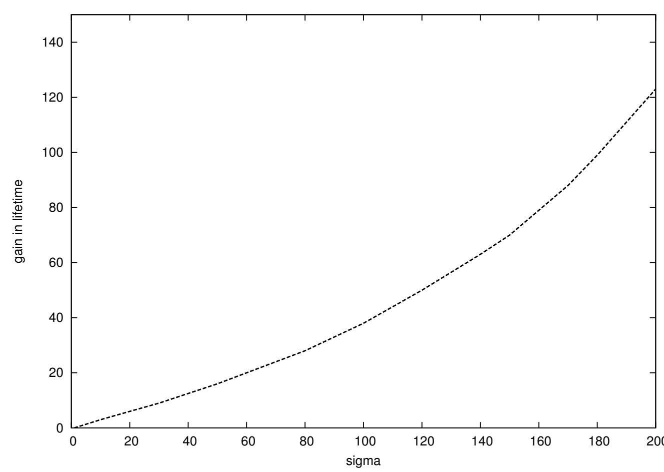


Fig. 10. Gain as function of $\sigma$ for $n = 8$

Another observation is that larger values of $\sigma$ could also result in a bigger gain in lifetime if exchange of subsystems between products in possible. In this plot we assumed 8 subsystems. In real practice the situation is a bit more complex. To make this work some products should be willing to donate parts while other products accept parts. This is what is called a donor-acceptor model. Observations like what the economical value of a subsystem is, should be taken into account. Also the cost of moving a subsystem from one product to another product could play an important role in this model. Let us consider some situations:

- If these $N$ products contain subsystems with large value for $\sigma$, meaning some parts will live long and other parts fail very quickly, the donor, acceptor approach will be very useful as shown in figure 10.
- Another observation that is easy to understand is: if these $N$ products contain one subsystem that will almost always be the first one to fail, the donor and acceptor approach will not help that much. This type of easily failing subsystems should be in stock as spare

parts.

To make this system work for distributed products, a way of communication and exchange appointments should be provided. In the next section a marketplace model is discussed.

### B. Negotiating between product agents

For the implementation of this exchange of parts concept, a Jade [2] multi-agent based system has been developed. The reason for using Jade are:

- this is a multi-agent-based system. Jade provides most of the requirements we need for our application like platform independence and inter agent communication;
- Jade is Java-based. Java is a versatile and powerful programming language;
- because Jade is Java-based it also has a low learning curve for Java programmers;
- agents can migrate, terminate or new agents can appear.

In figure 11 the architecture of the jade platform is depicted. The figure shows three so called containers. These containers can live on distributed platforms, but a network connection is obligatory. The Jade platform itself is in this figure surrounded by a dashed line. It consists of the following components:

- A main container with connections to remote containers
- A container table (CT) residing in the main container, which is the registry of the object references and transport addresses of all container nodes composing the platform;
- A global agent descriptor table (GADT), which is the registry of all agents present in the platform, including their status and location. This table resides in the main container and there are cached entries in the other containers;
- All containers have a local agent descriptor table (LADT), describing the local agents in the container;
- The main container also hosts two special agents AMS and DF, that provide the agent management and the yellow page service (Directory Facilitator) where agents can register their services or search for available services.

In our case we need only the main container for the product agents to run.

The product agent residing in a broken system can send the following information to a webserver:

- information about the status of the machine. What are the broken parts and what parts are still functioning;
- if available: information from the end-user if he or she prefers to be a donor, an acceptor or does not care;

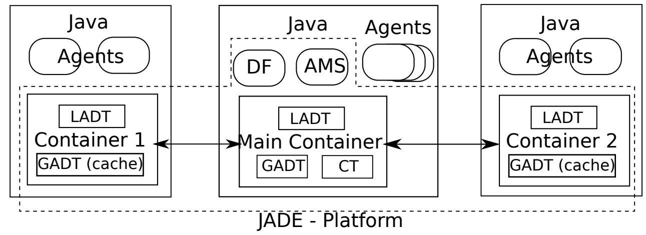


Fig. 11. Jade multi-agent platform

- the e-mail address of the end-user to contact the end-user to confirm the negotiation outcome;
- the maximum time to wait for a successful negotiation.

The webserver contains a Java servlet. After receiving the information from the remote product agent, this servlet will spawn a Jade agent equipped with the information from the original source in a Jade container residing in the computer system that runs the webserver. This situation is depicted in figure 12.

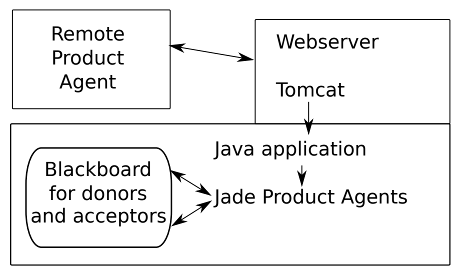


Fig. 12. Jade multi-agent platform

This Jade agent will take over the role of the product agent in the way that is described in section V-A. The Jade product agent of a broken product puts its id and status on a blackboard [9]. Using the same blackboard it will look for a situation where a complete product can be made. This will result in a set of product agents available for exchange. Now comes the question:

*To be or not to be a donor*

The product agent should decide to be a donor or an acceptor. This decision will also be influenced by the owner of the product. The following rules were applied to decide what the product agent should do:

- In the first place, the end-user of the product should decide what to do. If this user does not care, the following rules apply;
- the product with the largest amount of working subsystems should be the acceptor. If this does not result in a decision, use the next rule:
- if a part is broken with a large value of $\mu$ (expected lifetime is long) the role of acceptor is a good one, because of the fact that there is a big change that there exists a donor with this part available;

- if this still does not lead to a decision, a random choice is made.

When a match between two product agents is found, these agents both contact the end-user for confirmation. If both confirmations are positive, the negotiation is considered successful.

## VII. Recovering rare materials

When a product is manufactured, the product agent can build a list of materials used and also the position within the product where these materials can be found. In this way it is possible to recover materials using this list of ingredients. For example rare earth materials can be recovered and reused. Several ways exist to make this idea work:

- An on-board product agent can carry this information. In this situation the agent can also adjust the information to changes made to a product. If some parts are replaced during repair, the product agent will take care of updating the information.
- In case of an external product agent, the information will be available in the cloud. Adjusting this information to replaced parts during repair should be done as an extra step during repair.

The next problem to be solved is how to make the information available. Here we also consider two possibilities:

- The information can be presented in a human readable form by a web interface. The product itself could contain a small webserver that presents the information on demand. This same webserver could also be used to present a user manual or help in case of troubleshooting. In case of an external product agent, this agent can also be connected to a webserver for displaying the information.
- The information can also be presented in a machine readable form. In this case an XML-file will be made available so software can discover the structure and components of a certain product in combination with the materials used and their position within a product.

The concept of showing the places where interesting or expensive materials reside in a product has been implemented in a prototype for a discovery robot. This platform was enhanced with a product agent that would monitor the functioning of the different parts of the robot. This dynamic information was available using a small on-board web-server. This same webserver also showed information about the places where certain materials were used in building this robot. The product itself was thus carrying the information. A more detailed description can be found in [22].

## VIII. Related Work

In the field of agent-based production there are several important publications. Important work in this field has already been done. Paolucci and Sacile[25] give an extensive overview of what has been done. This work focuses on simulation as well as production scheduling and control. The main purpose to use agents in [25] is agile production and making complex production tasks possible by using a multi-agent system. Agents are also introduced to deliver a flexible and scalable alternative for MES for small production companies. The roles of the agents in this overview are quite diverse. In simulations agents play the role of active entities in the production. In production scheduling and control agents support or replace human operators. Agent technology is used in parts or subsystems of the manufacturing process. We on the contrary based the manufacturing process as a whole on agent technology and we have developed a production paradigm based on agent technology in combination with a production grid. This model uses only two types of agents and focuses on agile multiparallel production. The design and implementation of the production platforms and the idea to build a production grid can be found in Puik[26]. After production the product agents can be embedded, if possible, in the product itself. Agents for distribution, logistic applications and product manufacturing already exist [25]. In most situations agents represent human operators or negotiators. Jennings and Bussmann [7][17] introduce the concept of a product agent, in their terms workpiece agents, during the production. These agents do not however perform individual product logging and only play a role in the production phase. In our approach the product logging is the basis of the other roles of the product agent in other parts of the life cycle.

The use of a product is also studied by observing and/or interviewing end users [23] [24]. Some software applications do connect with their originating company to report the use by end users. Several proposals and implementations of including monitoring and documentation within the product itself are made and implemented. Burgess [4] [5] describes Cfengine that uses agent technology in monitoring computer systems and ICT network infrastructure. In Cfengine, agents will monitor the status and health of software parts of a complex network infrastructure. These agents are developed and introduced in the use phase of this infrastructure and focus on the condition of the software subsystems. In our approach this monitoring function for hardware and software is the role of the same agent that started as an agent during the manufacturing and already collected valuable information that can be useful to the end-user. Actually this agent is not necessarily the same software entity that played the role of product agent during production, but the belief base of the product agent is kept intact and handed over to a new incarnation of the product agent. Unlike Cfengine that is designed for computers and networks, our approach does not focus on certain types of products.

Gnedenko and Ushakov ([14] [15]) published standard works on reliability engineering using a firm mathematical basis. In this work the lifetime of products is a central issue based on the expected lifetimes of subsystems.

By using this same product agent again in the final phase of the life-cycle, component reuse and smart disassembly is a very important aspect when it comes to recycling of rare or expensive building material. Research in the field of recycling is overwhelming. Ellis [12] describes industrial methods to recycle rare earth elements. This article is about metallurgy and not about using information technology. Kovacs [19] proposes agent technology in car-recycling. This work focusses on exchange of information between enterprises that recycle and destruct used cars. There is however not a notion of a product agent in their approach. Another difference with our approach is that it focusses only on cars, while in our approach every good product deserves a product agent.

## IX. Conclusion

Adding an agent to a product that has knowledge of the product and the way it is made can help to extend the lifetime of a product. Besides it gives an opportunity to locate en reuse rare elements. These product agents are autonomous software entities that can assist in recycling.

In the concept that has been developed, agents play an important role in the whole life cycle of a product. This concept can be an enabling technology for the internet of things. The product agent will be the representative of a product. It is a software entity that collects information for a product from the internet, shares information with other product agents and sends information to other product agents using standard internet technology.

Interesting further research will be the insight in the reliability of subsystems. The distributed multi agent system of autonomous product agents can generate all kind of statistical interesting data about the MTBF of these subsystems. This could help manufactures to improve the quality of their products. It can also help in setting up a fair trading system for used spare parts.

## References


[1] Bloomberg. *Increase in prices of various resources over a period of 10 years*. 2009.

[2] N.R. Bordini, M. Dastani, J. Dix, and A. E. F. Seghrouchni. *Multi-agent Programming*. Springer, 2005.

[3] M.E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, Mass, 1987.

[4] M. Burgess. Cfengine as a component of computer immune-systems,. *Proceedings of the Norwegian Informatics Conference*, 1998.

[5] M. Burgess, H. Hagerud, S. Straumnes, and T. Reitan. Measuring system normality. *ACM Transactions on Computer Systems (TOCS) Volume 20 Issue 2*, pages 125–160, 2002.

[6] B. Burmeister, A. Haddadi, and G. Matylis. Application of multi-agent systems in trafc and transportation. *IEEE Proceedings on Software Engineering 144 (1)*, page 5160, 1997.

[7] S. Bussmann, N.R. Jennings, and M. Wooldridge. *Multiagent Systems for Manufacturing Control*. Springer-Verlag, Berlin Heidelberg, 2004.

[8] D. Cohen. Earth's natural wealth: an audit. *New Scientist*, (2605), 2007.

[9] D.D. Corkill, K.Q. Gallagher, and P.M. Johnson. Achieving flexibility, efficiency, and generality in blackboard architectures. *Proceedings of the National Conference on Artificial Intelligence*, pages 18–23, 1987.

[10] D.C. Dennett. *The Intentional Stance*. MIT Press, Cambridge, Mass, 1987.

[11] V. Dignum, J.J. Meyer, and H Weigand. Towards an organisational model for agent socienties using contracts. *Proc. of AAMAS*, (1), 2002.

[12] T.W. Ellis, F.A. Smith, and L.L. Jones. Methods and opportunities in the recycling of rare earth based materials. *The Metallurgical Society (TMS) conference on high performance composites*, (IS-M–796), 1994.

[13] L. Gasser. *Perspectives on Organisations in Multi-agent Systems*. Number LNAI 2086. 2001.

[14] B. Gnedenko, I.V. Pavlov, and I.A. Ushakov. *Statistical Reliability Engineering*. Wiley-Interscience, 1999.

[15] B. Gnedenko and I. A. Ushakov. *Probabilistic Reliability Engineering*. Wiley-Interscience, 1995.

[16] M. Gunther. The end of garbage. *Fortune*, 2007.

[17] N.R. Jennings and S. Bussman. Agent-based control system. *IEEE Control Systems Magazine*, (Vol 23 nr.3):61–74, 2003.

[18] J. Kletti. *Manufacturing Execution System - MES*. Springer-Verlag, Berlin Heidelberg, 2007.

[19] G. Kovacs and Heidegger G. Car-recycling sme network with agent-based solutions. *European Research Consortium for Informatics and Mathematics*, (73), 2008.

[20] L.J.M. van Moergestel, J.J.Ch. Meyer, E. Puik, and D.H. Telgen. Simulation of multiagent-based agile manufacturing. *CMD 2010 proceedings*, pages 23–27, 2010.

[21] L.J.M. van Moergestel, J.J.Ch. Meyer, E. Puik, and D.H. Telgen. Decentralized autonomous-agent-based infrastructure for agile multiparallel manufacturing. *ISADS 2011 proceedings*, pages 281–288, 2011.

[22] L.J.M. van Moergestel, E. Puik, D.H. Telgen, H. Folmer, M. Grünbauer, R. Proost, H. Veringa, and J.J.Ch Meyer. Monitoring agents in complex products enhancing a discovery robot with an agent for monitoring, maintenance and disaster prevention. *ICAART 2013 proceedings*, 2013.

[23] J. Nielsen and J. Levy. Measuring usability: preference vs. performance. *ACM*, 1994.

[24] J. Nielsen and R.L. Mack. *Usability Inspection Methods*. John Wiley & Sons, 1994.

[25] M. Paolucci and R. Sacile. *Agent-based manufacturing and control systems : new agile manufacturing solutions for achieving peak performance*. CRC Press, Boca Raton, Fla., 2005.

[26] E. Puik and L.J.M. van Moergestel. Agile multi-parallel micro manufacturing using a grid of equiplets. *IPAS 2010 proceedings*, pages 271–282, 2010.

[27] M. Wooldridge. *An Introduction to MultiAgent Systems, Second Edition*. Wiley, Sussex, UK, 2009.

[28] M. Wooldridge and N. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, (10(2)):115–152, 1995.