EXPERIENCE REUSE BETWEEN MOBILE PRODUCTION MODULES - AN ENABLER FOR THE FACTORY-IN-A-BOX CONCEPT

Erik Olsson[1], Mikael Hedelind[2], Mobyen Uddin Ahmed[1], Peter Funk[1]

[1]Mälardalen University, Department of Computer Science and Electronics,
P.O. Box 883, SE-721 23, Västerås, Sweden
[2] Mälardalen University, Department of Innovation, Design and Product Development
P.O. Box 325, SE-631 05, Eskilstuna, Sweden

{erik.m.olsson, mikael.hedelind, mobyen.ahmed, peter.funk}@mdh.se

Abstract: Today's increased demand for flexible and fast reconfiguration of production systems is seen as one of the key factors for survival by many branches, especially small and medium sized enterprises. To enable adaptable and flexible production, we propose an integrated experience reuse system assisting in setup, operation, maintenance and repair. We present three subsystems that facilitate experience reuse between different engineers and operators working with standardised production modules. It is composed of three separate software components enabling: a) easy programming and control of robot cells, b) monitoring and condition based maintenance, c) distributed experience reuse. The results presented in this paper have been developed within the Factory-in-a-Box project, the ExAct project and the Eken project.

Keywords: Mobile Production Units, Automated Production, Experience Reuse, Decision Support Systems, Condition Monitoring, Intelligent Agents, Case-Based Reasoning, Experience Sharing Forum, Cell Programming, Cell Control

1. INTRODUCTION

Traditional manufacturing industry is optimized towards large volume production. The demand for handle large variations in production volumes and fast reconfiguration to enable production of different products in the same production line is increasing. This ability is by many seen as a key factor for survival in an increasingly global market. Especially small and medium sized enterprises (SMEs) have a hard time meeting fluctuating production volumes originating in low-batches and short time-horizons on orders, both trends on the market today. Flexibility and re-configurability are concepts that are becoming increasingly important in order to be competitive (Hedelind and Jackson, 2007). There is an increasing interest in production capacity/modules that in short time and without extensive in house expertise can be configured and enable increased production capacity during shorter periods.

One proposed solution is to lease production capacity in the shape of production modules. Ideally they arrive to the production site configured for the current production needs and ready to start with a minimum set-up time. Studies also show that companies need to compete, not only with cost, quality, delivery reliability and flexibility, but other more innovative concepts and competitive factors may have to be developed (Winroth, 2007). Such competitive factors may e.g. be mobility, which enables companies to reuse the same hardware and software solutions on several locations, adding production capacity where it is needed at the time.

The Factory-in-a-Box project is a research initiative in the area of mobile production modules (Jackson and Zaman, 2006). The factory-in-a-Box project and the underlying research projects has resulted in a wide range of innovative enablers and concepts have been explored and implemented in a number of demonstrators and of which some already found their way into production industry, e.g. (Hedelind et al., 2006) is now integrated in an industrial partners production line. The aim of the project is to research and develop mobile production modules that can be moved between production sites e.g. in order to meet with production volume peaks. Factory-in-a-Box is a three-year research project that started 2005 and involves four different universities and several industrial partners. The concept of the project is built on a conceptual business case where a fictitious company leases production modules to companies in need of temporary production capacity. The modules are automated to some extent and can be shipped all over the world in order to enable companies to change their production capacities to meet with customer needs.

One of the difficulties with leasing automated production modules to companies is the experience and knowledge needed by the customer in order to start-up and use the production module. The need to send an experienced person with the production module in order to operate the module is adding costs and requirements on the leasing contract.

In order to realize the Factory-in-a-Box concept, several enablers have been researched in the project. This paper will present a combination of three of the enablers that together can meet with the need for expertise knowledge at the production site.

In order to distribute the production modules to different companies, which are often unskilled in the concept and area of automation, there is a need for flexible cell programming and control. This area has been researched within the Factory-in-a-Box project and resulted in a robotic working cell configuration tool called *Cell Configurator*. The Cell Configurator is reconfigurable cell-pc software that enables quick and easy reconfiguration of the program logic in an automated manufacturing cell. It also provides means for scheduling maintenance and error recovery where the operator is guided through the process of maintaining and recovering the cell.

Several steps in the life cycle of a Factory-in-a-Box can benefit from a systematic approach for experience reuse. For fast, flexible and mobile configuration and operation of a Factory-in-a-Box production module, a source of experience is an invaluable asset. By systematically collecting experience from different Factory-in-a-Box solutions - successful cases as well as unsuccessful ones, one will get a continuously growing case-base of experience that will provide the Factory-in-a-Box company with a valuable platform for decision support. In this paper we propose a case-based approach and tool to facilitate a systematic collection and reuse of experience during the configuration and operational phase of a mobile production module's life-cycle.

The configuration software, combined with the reuse of experience and distributed case-base solution, will provide a valuable asset if the leasing business case is to be realized. The three systems will be described in the paper, and will together present a novel and powerful enabler for the Factory-in-a-Box concept. Section 2, following this section, will present an overview of the system presented in this paper. Section 3 describes the Factory-in-a-Box project. Section 4 presents the Cell Configurator software and how this software may be used to collect experience and data from the Factory-in-a-Box production modules. Section 5 describes how experience-reuse can be realized through case-based reasoning. Section 5 presents the maintenance agent that is used to monitor the current state of the production cell. Section 6 presents how case-based reasoning can be used for incorporating experience reuse. Section 7 will present a discussion and some conclusions.

2. PROPOSED SYSTEM OVERVIEW

In this chapter we focus on the different kinds of experience needed to successfully set up and run a mobile production unit (Jackson and Zaman, 2006). When a production unit is ordered it first needs to be configured for the current production needs so it can be ready to start within a minimum of setup time when arriving at its destined location. Knowing about the customer needs and demands, a basic setup is first performed involving a feasible configuration of the structure and layout of the production unit. Secondly, the behaviour of the robots, machines and other I/O-controlled equipment is programmed according to meet these demands. Configuration/reconfiguration and programming is done by the configurator part of the Cell Configurator which is a tool for easy and flexible configuration and programming of mobile production units. After successful setup and integration of the production unit, the production process can be started. In production mode the controller part of the cell configurator controls the production process. During production the unit is constantly monitored by a maintenance agent (Funk et al., 2006). The task of the maintenance agent is to report any anomalies occurring during production e.g. immediately shut down the process and report to a technician if a robot is loosing its grip on an object during assembly or if some machine or robot breaks down. Different failure codes are reported according to the severness of the failure (see Fig 1). An experience sharing interface is always present in the unit working as a general knowledge and experience sharing agent. It has access to a local and/or distributed database containing previously saved cases of experience from other mobile production units. Except from the general experience located in the experience sharing agent experience is also saved in the form of:

1. Production unit configurations with accompanying controller programs suitable for different production scenarios and demands.
2. Fault and maintenance libraries describing symptoms, diagnosis, actions, prognosis etc of various failure modes that can occur.

3. FLEXIBLE, MOBILE AND QUICK PRODUCTION MODULES

The Factory-in-a-Box project was started around the vision of having a fictional company that could lease out production capacity on demand[1]:

> *"The Factory-in-a-Box concept will consist of standardized production modules that are installed in a container and transported by e.g. a truck or by train. The modules may then rapidly be combined into production systems that can be reconfigured for a new product and/or scaled to handle new volumes. Production capacity may be provided as a mobile and flexible resource that rapidly can be tailored to fit the needs of a company, at a specific point of time."*

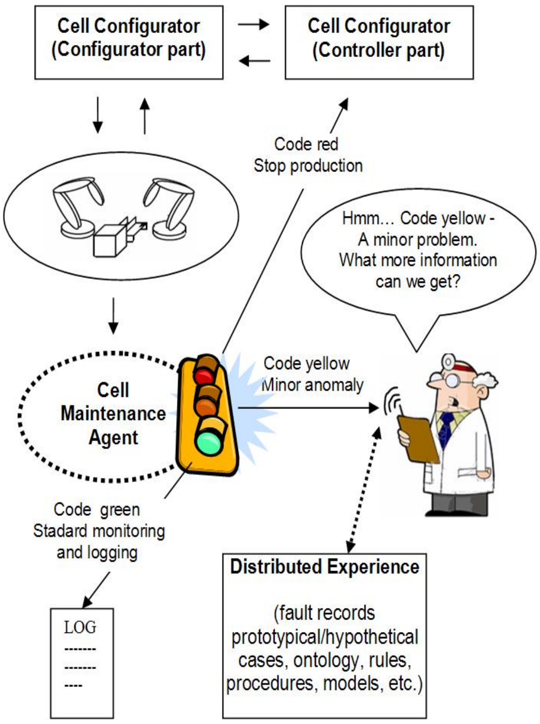

Fig. 1. A mobile production unit scenario demonstrating operation of the three systems.

[1] Quote from the Factory-in-a-Box project plan, 2005-02-09

The business concept discussed within the Factory-in-a-Box project implies that there should exist some company, which has a set of generic production modules that can be configured to suite the needs of the company leasing the cell. The company in need of production capacity may then specify the production capacity they need and the Factory-in-a-Box company designs, configures and puts together a production cell that matches the leasing company's needs. Another application of the mobile production modules could be that a company develops or buys this kind of production module and move it between their different production plants in order to handle local production peaks. In both applications, there is a need for knowledge and experience at the production site to handle set-up, production, maintenance, and failure recovery."

3.1 Enabling Features

Three keywords were chosen within the project as enablers for this vision: *flexibility*, *mobility* and *speed*. If these three keywords, or features, are realized within each production module; the vision of standardized, mobile production modules could be realized. The different features are associated with special requirements that are transferred to the manufacturing cells. For example, if the cell is to be mobile, there are special requirements on how the cell may be transported, or if the cell is to be flexible there may be special requirements on how flexible.

This paper is focusing on the software used for controlling this kind of automated and mobile production cell, while providing the operator with support. This means that the major requirements discussed in this paper comes from the flexibility aspects of the production cell when it is to be reconfigured and also from the fact that the mobility feature leads to that there may be operators using the cell which have no real experience from this kind of production cells.

3.2 Flexibility and Reconfigurability

There is a great need for flexible and reconfigurable production systems that can handle changes in e.g. production volume or product variants (Hedelind and Jackson, 2007). There are several definitions and classifications of flexibility and reconfigurability being used today. In this paper flexibility of a production system will be defined and determined by the systems sensitivity to changes (Chryssolouris, 1996). As there are many different types of flexibility it is possible to classify flexibility into different categories. Slack (1987) found out through his study that four different types of flexibility was considered important by the companies that took part in the study: (1) product flexibility, (2) mix flexibility, (3) Volume flexibility, and (4) delivery flexibility. However, there exist several other classifications of flexibility e.g. the eight different types of flexibility that Browne et al. (1984) defined. A flexible production system is, using the definition in (Jackson 2000): a system that has been designed in accordance with the ability to deal with changes effectively, without that any development or change of the actual system is needed.

Reconfigurability on the other hand is defined as the systems ability to adapt to changes, meaning that the system can be changed in a quick or effortless way to handle different changes. ElMaraghy (2005) defines a reconfigurable manufacturing system as a production system that promises customised flexibility on demand in a short time, in comparison to a flexible manufacturing system that provides generalized flexibility designed for the anticipated variations and built-in a priori.

Both flexibility and reconfigurability is important when designing production systems. When designing each cell in a production system, it is important to consider how different changes will be handled, and which changes that are handled by the system as it is, and which changes that leads to implementations of changes in the system. This becomes increasingly important if the production cell is capital-intense, such as robotic working cells. When designing such cells it is even more important to build them in such a manner so that they will pay off over time, without to much overhead costs originating in the implementation of changes.

4. CELL PROGRAMMING AND CONTROL

The background of the Cell Configurator is the need for a visual and intuitive way to program and control robotic working cells. One of the demonstrators built within the Factory-in-a-Box project contains a pair of robots that cooperate when assembling a cabinet. Another robot is to be added later on into the cell. The cell also, along with the robots, contains several other equipments and fixtures e.g. conveyors, gluing station etc, that is controlled by I/O signals.

One of the main priorities when the production module was designed was its ability to be flexible enough to handle all available variants of the product which was produced, and also its ability to be easily changed to match future versions of the product. This called for a reconfigurable robotic production cell where it should be easy to add new fixtures, grippers and other equipment.

> As ElMaraghy (2005, pp. 262) points out:
> *"do most hardware reconfiguration require major changes in the software used to control individual machines, complete cells, and systems as well as to plan and control the individual processes and production."*

A survey of available software solutions was conducted to find a suitable solution for programming and control of the cell. However, no suitable solution was found that matched the requirements of the production module.

Thus, the development of a prototype of a software tool that met with all the requirements was started. The main requirements that were placed on the software tool included that it should: (1) allow the user set-up the structure and layout of the working cell, (2) allow the user configure the application to show the status of all the equipment in the working cell, (3) allow the user to program the behaviour of the robots and other I/O-controlled equipment in the working cell through a visual and intuitive user interface, (4) allow the user to connect to other surrounding IT-structures or special equipments such as barcode readers through well defined interfaces, (5) control the production process in the working cell during production using the program the user has created, (6) display the current status of the production to the user, (7) guide the user when recovering the cell from any production failures, and (8) allow the user to set-up and execute maintenance plans for the cell.

The software, Cell Configurator, was developed over two years and the prototype was tested and evaluated during the summer of 2006. As the result of the evaluation was promising, a decision was made to make further developments in order to set the software into production. Today, the software is used within the production site of the industrial partner in order to program, control and monitor the production process of the robotic working cell.
The software developed contains three parts: the *Cell Designer*, the *Sequence programmer*, and the *Production window*. The Cell Designer is used to create a layout of the cell using drag-and-drop techniques. It also allows the user to configure some of the properties of the different equipments that are added to the cell, such as I/O signals and variables. The Sequence programmer is used to program the behaviour of the cell using an icon-based flowchart programming style. The Production window is used during production to allow the user to start and stop production, as well

as monitor the production process, see statistics and perform maintenance plans.

One of the features of the Cell Configurator is its ability to monitor production and recognize if the production has entered an abnormal state. This is done by monitoring variable values and process states that are defined by the programmer. If production is halted due to such an abnormal state, or if any other failure happens that e.g. leads to an emergency stop, the Cell Configurator will stop production and enter a failure mode. As production is entering the failure mode the Cell Configurator will create a snapshot of the current state of production. The snapshot contains the current state of all the variables, signals and processes in the system. This snapshot will then be included in a failure report that also contains some log-history and some other information about the current production line and project. This report is then saved for future reference and the Cell Configurator will display information to the operator in the form of a *wizard* that instructs the operator in how to solve the current problem.

5. THE MAINTENANCE AGENT

A maintenance agent is specialized in interpreting data from the device it is connected to. The agent observes its environment through one or more sensors. Additional information about the environment may also be acquired through communication with other agents or systems. The agent may have some basic domain knowledge about when to bring the findings to the attention of a human and when to shut down a process. The agent also has social skills to communicate its findings. It may also ask for additional information to make a final decision and it has facilities to receive appropriate feedback (Funk et al., 2006).
Handling groups of sensors with a dependency between measurements enabling sensor agents to collaborate and learn from experience, resulting in more reliable performance.

Maintenance agents may also improve their performance, e.g. recalibrate sensors if needed, or determine if sensors are faulty. Similar sensors may also share experience enabling them to avoid repetition of similar failures or make estimates on their reliability. The agent has a perception module containing necessary information for basic pre-processing and abstraction of the sensor data. It is a learning agent (Russell and Norvig, 2002) with a deliberation module containing a memory that stores basic domain knowledge and capabilities to make a decision using e.g. Case-Based Reasoning, Rule-Based Reasoning etc.

An example of a local Case-Based maintenance agent is given in (Olsson et al., 2004). The agent uses a Case-Based Reasoning method and a nearest neighbour approach for a light weight solution of recognizing and diagnosing audible faults on industrial robots (see Fig. 2). Sensor signals such as sound is recorded and compared with previous recordings, although also currents (Olsson, 2007) and other input signals have been explored and works appropriate.

6. EFFICIENT KNOWLEDGE AND EXPERIENCE SHARING USING CASE-BASED REASONING

Human experience is a valuable asset and could be even more valuable if artificially stored and reused in an efficient way. Engineers/technicians have experience which may have been collected during many years both from successful solutions as well as from very costly mistakes. As an example, when a less experienced technican/engineer is confronted with a new problem she/he might start to analyze the whole situation and try to find the source of the problem by using a manual, own and others experience, and other sources. This may be a very time-consuming task and may sometimes result in not finding a proper solution to the problem. In that case she/he needs to find other sources for help and a very common way is to ask his/her senior engineers who have more experience. A more experienced person might start to think by himself: "Have I ever faced any similar problems, and in that case, what was that solution?" and refer the problem with her/his past solution to the less-experience person. The less-experienced person then solve the problem and learn the new experience and store it in her/his own memory for future use. These kinds of experiences are a valuable asset for the companies both in the presence and absence of the experienced person. It is also possible to save a large amount of time and money if those experiences could be captured and stored in such a way that those can be reused in the future and shared in the collaborative companies. Such kind of human thinking, intelligence and reasoning-models can be found in case-bases reasoning (CBR) methodology. The following text describes about CBR and how CBR can be used for experience reusing.

6.1 Case-Based Reasoning (CBR)

Solving new problems with experience from earlier cases is a model of case-based reasoning and it is a very similar approach compared to human problem-solving behaviour. Aamodt and Plaza introduce a life-cycle of CBR (Aamodt and Plaza, 1994) with four main steps: Retrieve, Reuse, Revise, and Retain for implementation of such cognitive model.

The CBR methodology is often used to solve new problems based on learning from similar cases (i.e. existing experience) stored in a case library that is obtained by storing previous similar situations (Watson,. 1997). Figure 3 shows the CBR cycle according to Aamodt and Plaza (1994).

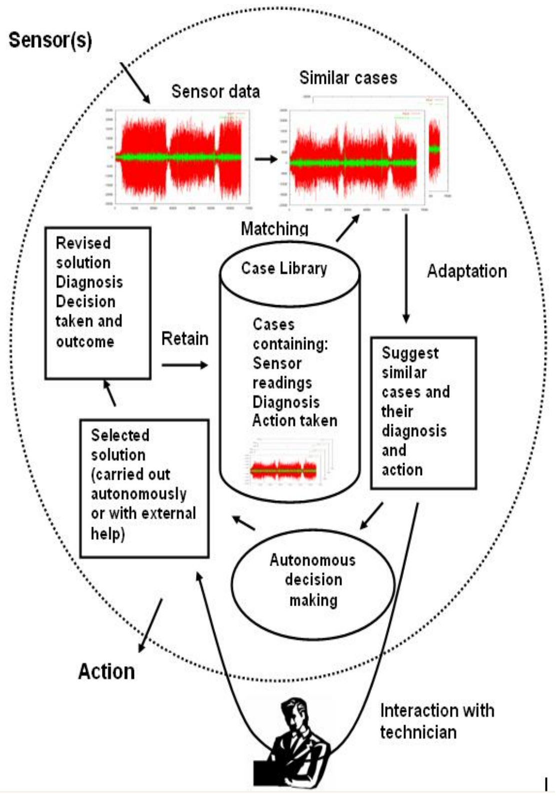

Fig. 2. Outline of a cell maintenance agent in its environment.

Three different kinds of approaches are mainly used in commercial CBR systems; such as textual, conversational and structural, but all of them generally follow the same CBR cycle that depicted in figure 3 (Bergman et al., 2003).

6.2 Knowledge and Experience Sharing

One project that is focusing on experience reuse and sharing is the EKEN project (Efficient knowledge and experience reuse within the business world). Eken is a knowledge and experience sharing management tools using CBR and other artificial intelligence techniques and methods. The result of the project is aiming to provide facility to share knowledge and identify people and their experiences with similar tasks and problems at different companies. An easy web-based tool has been implemented to access, collect, reuse, and spread experience and knowledge, both in-house and in public environments on the basis of needs. A simple overview of the knowledge and experience sharing system and how it works is shown in figure 4.

The users of the system gains access to three different approaches for sharing experience, as shown in figure 4: general experience sharing, domain specific experience sharing, and a forum for experience sharing. The following sub-sections illustrate these different approaches of the system.

General experience sharing

This is an approach which will work when experiences are more general, simple and without need to be dependent on any specific knowledge domain. Experiences might be structured or semi-structured but fully textual-based and the system provides the service of sharing the users' own experience in natural language. This approach has the ability to adapt to different circumstances and scenarios to make the structure of the experience easy and flexible.

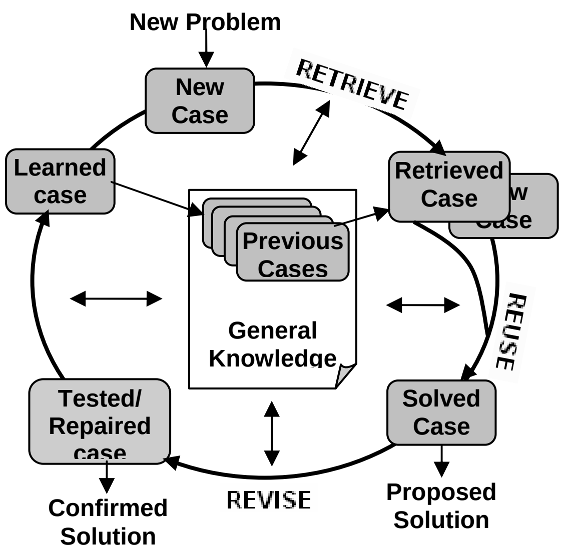

Fig. 3. CBR cycle according to Aamodt and Plaza. (Aamodt and Plaza, 1994)

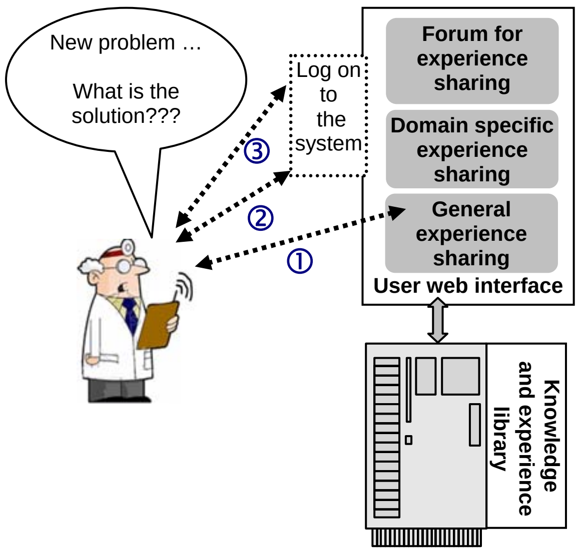

Fig. 4. An overview on how the experience sharing system works.

Using this approach the user may structure her/his own experience as she/he sees suiting for the problem at hand, e.g. some user can structure their experience starting from the problem, explaining the solution and then place recommendations, others can have a question and anwer approach. The system has different criteria to symbolize users' experience in a structured or a semi-structured way but all the knowledge is represented through natural language i.e. in free text which will increase the user acceptance. In (Ahmed et al., 2007b) the system is described in detail; its functionality, pros and cons of the knowledge and experience reuse.

A domain specific experience sharing

This approach is work well when the domain is specific and the experiences are a mixture of some pre-extract features with their corresponding values, either in symbolic or numerical nature, as well as some context or contextual information. This approach is usually represented in a natural language. The hybridization of symbolic or numerical data along with textual data enables new possibilities of utilizing contextual awareness in a CBR system, leading to more reliable and efficient experience reuse. This system also different security levels between the comparative users as well as competitive users; experiences can be encapsulated form the incorrect/false users. Depending on the user and their security level; system will allow sharing knowledge and reusing experience among the collaborating companies. Since the experiences have the authors' information, the system also helps in identifying the right person, e.g. there may be an engineer or operator near by and available for personal guidence, this would be an ultimate solution. More information about the system and how it works can be found in (Ahmed et al, 2007b).

Forum for experience sharing

This is a more commonly used approach to share, discuss and come in contact with experienced persons with valuable knowledge. However, the main disadvantage to the other approaches is how to identify and retreive relevant information in an easy way. The Eken-system works with user's profiles which is updated automaticall and has options to set user's preferences according to their interests and incorporates an intelligent mailing notification system. These are functions that helps the user to find the knowledge and experience that she/he is looking for. It is an intelligent forum that reduces number of unnecessary information and avoids to store redundant information. For example, when a user post a new topic, the system will search the similar topics from the whole database and send a list of similar topics along with the similarity value to the moderator of the system. The moderator will revise this and if it seems very close to the previous ones then he will make a request notification to the user to use the pervious one to join in the earlier discussion. Information are structured similar to other systems, for example the user can make comments, gives positive and negative feedback and asking questions and give answers on each topics separately, which will make searching easy. There is also functionality for the user to make reports on topics or request a removal of any topics, in cases where there are e.g. any ethical issues.

7. DISCUSSION AND CONCLUSIONS

This paper has described three different technical solutions that have been implemented and tested individually. The purpose of this work has been to design a system where the three technologies has been merged into one system that enables operators with no or small experience of automation technologies to operate a robotized production module with help of a control software that incorporates experience reuse over the internet. The software system also provides the operator with help to recover the working cell after that an unexpected halt has occurred.

The three sub-systems; Cell Configurator, Monitor Agent and experience reuse server, has been developed in different research projects; Factory-in-a-Box, ExAct and Eken respectively. The system described in this paper is intended as an enabler for distributed production modules, and for the Factory-in-a-Box business concept.
The different sub-systems provide different enablers for operators that are to work with the mobile production modules:

Cell Configurator:

- Intuitive graphical user interface
- Drag-and-drop icon-based programming interface
- Methods and techniques for cell control
- Systematic error and fault recovery system

Monitoring agent:

- Intelligent/learning sensoring system
- Ability to predict possible problems before a failure occurs

Experiences reuse system:

- Ability to share experience over a network
- Ability to reuse previous experience to solve current problems

Those systems combined could prove to be the innovative enabler that can provide sufficient operator support so that event a more inexperienced operator may operate a Factory-in-a-Box module.

7.1 Failure recovery scenario

Suppose we have a mobile production module that is leased to a SME for manufacturing of some part. The module contains an industrial robot and a lath that the robot is tending. At one point do one of the gear-boxes in the robot start to malfunction in some way. One of the monitoring agents will pick up an unclassified sound signature from the robot as it moves via a wireless Bluetooth microphone placed on the robot. The agent tries to classify the sound signature through its case library but fail to do so. The agent will alarm the Cell Configurator that the robot is in danger of failure and pass the sound signature along with the alarm.

The Cell Configurator will check the status of the cell and see that production is currently running as it is supposed to, so the system performs a “soft stop”, i.e. the robot may finish what it is doing but then run into a halt.

As the production has halted the Cell Configurator will enter *failure mode* and create a *failure report*. The failure report will include a snapshot of the current state of the production module, the sound signature from the agent, some log-history and other information about the currently run project.
As the failure report is finished the Cell Configurator will connect to an online case-server and send the report to the server. When the server receives the report it will extract the relevant information from the report and create a new *problem case*. The problem case will be sent into the case-base where it will be matched against previous cases and classified according to earlier experience. If a good match of the sound signature is identified the server will send user instructions back to the Cell Configurator, containing information on how the operator can proceed from the current situation. The server will also alert the “Factory-in-a-Box company” that one of the robots leased out to the specific company has a gear box that is in need of maintenance.

The Cell Configurator will display the information to the operator on-site and guide him/her through the process of restarting production. The Cell Configurator may also provide information to the monitoring agent, so that the next time the agent records the same sound signature, it will be able to classify the signature and report more information to the system.

7.2 Other cases of experience reuse and sharing

Experience sharing and reuse may be applied in more cases than just the one presented in the scenario in section 7.1. One could also consider experience/knowledge reuse for the actual configuration of the production module, matching earlier hardware and software solution to the current production needs. Another possibility is the forum for experience sharing where the operators may come in contact with earlier cases and more experienced persons that may provide help and guidance over the internet.

7.3 Future work

This paper has presented a conceptual solution built on three different systems that have been developed and tested separately. One likely future task would be to actually implement the complete system together, creating interfaces between the different sub-systems. This could result in a system that could be tested in future projects where demonstrators are developed. The Factory-in-a-Box project will have one last demonstration at the Technical Fairs in Stockholm during October 2007; this could provide a good opportunity for such a demonstration. This opportunity will be investigated and a first prototype of the system will be developed if time and resources allow for it.

ACKNOWLEDGEMENTS

Factory-in-a-Box is a joint project between 4 universities and more than 10 multinational and national companies, see www.factoryinabox.se, the project is funded by participating industrial partners and the Swedish Foundation for Strategic Research.
ExAct is a 3 year project exploring methods and tools for experience reuse in manufacturing industry, see www.idt.mdh.se/exact, the project is funded by participating industrial partners and the Swedish Foundation for Strategic Research.

Eken is a one year project where tools for generic experience reuse between small and medium sized companies, http://www.erfarenhetspoolen.se/. The project is funded by *EU mål 2* and *Västmanlands landsting*

REFERENCES

Aamodt, A. and E. Plaza (1994). Case-based reasoning: foundational issues, methodological variations, and system approaches, *AI Communications*, 7 (1), pp. 39-59.

Ahmed, M. U., E. Olsson, P. Funk and N. Xiong (2007a). Efficient Condition Monitoring and Diagnosis Using a Case-Based Experience Sharing System. 20th International Congress and Exhibition on Condition Monitoring and Diagnostics Engineering Management, COMADEM 2007, Faro, Portugal.

Ahmed, M. U., E. Olsson, N. Xiong and P. Funk (2007b). A Case-Based Reasoning System for Knowledge and Experience Reuse. 24th annual workshop of the Swedish Artificial Intelligence Society, Borås, Sweden,.

Bengtsson M., E. Olsson and P. Funk (2004). Technical Design of Condition Based Maintenance Systems - A Case Study using Sound Analysis and Case-Based Reasoning. In MARCON (Maintenance and Reliability Conference), pages 12, Knoxville, TN, USA, May 2004.

Bergman, R., K.D. Althoff, S. Breen, M. Göker, M. Manago, R. Traphöner and S. Wess (2003). Developing Industrial Case-Based Reasoning Applications-The INRECA Methodology. Carbonell J. G. and Siekmann J., Lecture Notes in Artificial Intelligence, Springer.

Browne, J., K. Rathmill, S.P. Sethi, KE. Stecke (1984). Classification of flexible manufacturing systems. *The FMS Magazine* April 114-117

Chryssolouris, G. (1996) Flexibility and its Measurement. Annals of the CIRP, Keynote Paper, 45:581-587

Funk, P., E. Olsson, M. Bengtsson and N. Xoing (2006). Case-Based Experience Reuse and Agents for Efficient Health Monitoring, Preventive and Corrective Actions. Proceedings of the 19th International Congress on Condition Monitoring and Diagnostic Engineering Management, COMADEM 2006, pp. 445-453, Luleå, Sweden, Editor(s):Kumar, U., Parida A., Rau R.B.K.N.

Elfving S. and M. Jackson (2005) A Model for Evaluating and Improving Collaborative Produicvt Development. International Conference on Engineering design. ICED'05. August 15-18, Melbourne, Australia.

Elfving S., P. Funk (2006). Enabling Knowledge Transfer in Product Development and Production – Methods and Techniques from Artificial Intelligence. In The 1st Nordic Conference on Product Lifecycle Management, pages 13, Gothenburgh, January.

ElMaraghy, H. (2005) Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems*, Vol. 17 pp.: 261-276

Hedelind, M., Jackson, M. (2007) The need for Reconfigurable Robotic Systems. To be published in Proceedings of the CARV'07 conference, July 22-24, Toronto, Canada

Hedelind, M., P. Funk and M. Milic (2006). Intelligent Buffer Storage System – Enabling Fast and Flexible Assembling with Industrial Robots, *Journal of Intelligent & Fuzzy Systems*, pages 8, IOS Press.

Jackson, M. (2000). An Analysis of Flexible and Reconfigurable Production Systems – An Approach to a Holistic Method for the Development of Flexibility and Reconfigurability. Dissertation No. 640 Linköping University

Jackson, M. and A. Zaman (2006). Factory-in-a-Box – Mobile Production Capacity on Demand. Proceedings of the 2006 IJME – INTERTECH Conference October 12-21 New York City USA

Olsson, E., (2007). Identifying Discriminating Features in Time Series Data for Diagnostics of Industrial Machines. The 24th annual workshop of the Swedish Artificial Intelligence Society, Borås, Sweden, May, 2007.

Olsson, E., P. Funk and N. Xiong (2004). Fault Diagnosis in Industry Using Sensor Readings and Case-Based Reasoning, *Journal of Intelligent & Fuzzy Systems*, Vol. 15, ISSN 1064-1, p10, IOS Press, December, 2004.

Russell, S.J. and P. Norvig (2002). Artificial Intelligence: A Modern Approach (2nd Edition). Prentice Hall; 2nd edition (December 20, 2002), ISBN : 0137903952

Slack, N. (1987). The flexibility of manufacturing systems. *International journal of Operations & Production Management* 7:35-45

Watson, I. (1997) Applying Case-Based Reasoning: Techniques for Enterprise Systems, Morgan Kaufmann Publishers Inc, 340 Pine St, 6th floor, San Fransisco, CA 94104, USA.

Winroth, M., M. Jackson (2007). Manufacturing Competition through the Factory in a Box Concept. In proceedings of the POMS 18th Annual Conference May 4-7 Texas USA