Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 36 (2009) 93–101
Expert Systems with Applications
www.elsevier.com/locate/eswa

Knowledge acquisition for decision support systems on an electronic assembly line

Sébastien Gebus *, Kauko Leiviskä

Control Engineering Laboratory, Department of Process and Environmental Engineering, University of Oulu, P.O. Box 4300, FIN-90014 Oulu, Finland

Abstract

Increasing global competition has made many manufacturing companies recognize that competitive manufacturing in terms of low cost and high quality is crucial for success. Real-time process control and production optimization are, however, extremely challenging areas because manufacturing processes are getting ever more complex and involve many different parameters. This is a major problem when building decision support systems especially in electronics manufacturing. Although problem-solving is a knowledge intensive activity undertaken by people on the production floor, it is quite common to have large databases and run blindly feature extraction and data mining methods. Performance of these methods could, however, be drastically increased when combined with knowledge or expertise of the process.

This paper describes how defect-related knowledge on an electronic assembly line can be integrated in the decision making process at an operational and organizational level. It focuses in particular on the efficient acquisition of shallow knowledge concerning everyday human interventions on the production lines, as well as on the factory-wide sharing of the resulting information for an improved defect management. Software with dedicated interfaces has been developed using a knowledge representation that supports portability and flexibility of the system. Semi-automatic knowledge acquisition from the production floor and generation of comprehensive reports for the quality department resulted in an improvement of the usability, usage, and usefulness of the decision support system.

Keywords: Decision support system; Knowledge acquisition; Quality management; Optimization; Control

1. Introduction

Industrial processes are subject to many known or unknown malfunctions during their operational lifetime leading to a reduction in their efficiency. Problem-solving is therefore an essential task for corporations that seek to improve their profitability. Consequently, process control needs much attention as it aims at preventing or reducing the effects of these malfunctions. Most control methods have, however, something in common; they manipulate data and parameters in order to take decisions based upon the estimated behavior of the system. This is especially true in the electronics industry, where the rapidly evolving environment and short life cycle of products do not allow the extensive build-up of operational experience on the factory floor. As a result, quality-related decisions rely heavily on measurement data that focuses often on product rather than on process characteristics. The problem with data-based approaches, however, is that data does not exist naturally in a factory; it has to be collected, stored, prepared, and eventually mined. Data might also be incomplete, inaccurate, or simply irrelevant to the problem that is being investigated. Moreover, problem-solving is a complex process based upon the subjective and knowledge intensive evaluation of the situation and leads to actions with uncertain effects. Without relevant information about the situation workers are unable to choose an appropriate course of action to solve a problem. These difficulties might be overcome by taking knowledge about the environment, the

* Corresponding author.
E-mail address: sebastien.gebus@oulu.fi (S. Gebus).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.09.058

task, and the user into consideration during the data analysis (Gebus, 2006).

Knowledge refers to what one knows and understands. It is "meaningful links people make in their minds between information and its application in action in a specific setting" (Dixon, 2000). In the industry, it refers to the sum of information relevant to a certain job and usually, we get things done successfully by knowing an answer or how to find an answer, or knowing someone who can. In this context, an expert is a person with special or superior skills or experience in the particular area and who is widely recognized as a reliable source of knowledge in that area. For example, an operator with years of experience in quality control on electronics assembly lines would be widely recognized as having extensive expertise in recognizing defective soldering. Ensuring that this accumulated experience does not leave the organization is a problem taken more and more seriously as many companies prepare for the retirement of the "baby boom" generation and face potential mass departure of valuable staff. In the electronic industry, and more specifically in the IT industry, this problem is not as acute as elsewhere as the average age of employees is lower than in more mature industrial sectors. Nevertheless, the IT industry is not immuned to the problem. Cost cutting measures often lead to production delocalization and the consequent loss of knowledge and expertise in industrial countries is not compensated as people and their know-how seldom follow the delocalization process. For these reasons, more attention should be given to the broader concept of decision support systems (DSS).

A DSS can be defined as "an interactive, flexible, and adaptable computer-based information system, especially developed for supporting the solution of a non-structured management problem for improved decision making. It utilizes data, provides an easy-to-use interface, and allows for the decision maker's own insights" (Turban, 1995). This definition is interesting as it emphasizes the idea that a DSS is mainly a framework in which problem-solving takes place and data analysis can be implemented efficiently along with other methods. Moreover, the development of knowledge-based DSS is justified by the inability of decision makers to efficiently diagnose based upon measurement data many malfunctions, which arise at machine, cell, and entire system levels during manufacturing operations (Özbayrak & Bell, 2003). In this context, knowledge-based approach takes the advantage of the fact that the people operating the process most likely have the best ideas for its improvement. The integration of these ideas into the problem-solving approach leads to the solution for the long term process improvement (Seabra Lopes & Camarinha-Matos, 1995). Additionally, as the use of knowledge and more generally qualitative information better explains the relationships between input process settings and output response, knowledge integration well indicates the improvement in the understanding and usability of DSS (Spanos & Chen, 1997). For a DSS to be efficient, it should therefore not rely only on one single type of input, nor should it rely on one single source.

In the rest of this paper, we shall examine the possibility to integrate knowledge in general, and especially shallow knowledge, into the framework of decision making process. Section 2 presents the problems related to knowledge acquisition and knowledge-related improvements in human/machine interactions. Section 3 presents our contribution to that field through a case study that is followed by the discussion on the results and the conclusion.

2. Knowledge acquisition

Knowledge exists in any organization and can take various forms. Nevertheless, companies have difficulties to create systems for acquiring, retaining, and accessing it, especially in the case of highly specialized knowledge. When this knowledge belongs to a physical person, it is also referred to as expertise or experience. Sometimes, however, knowledge resides also within non-human objects either in a straight and formal way (e.g. expert systems), or in a conceptual way. Design information, for example, can be seen as an expression of knowledgeable choices made by the designer (Gebus & Juuso, 2002). In other words, knowledge exists as soon as human interaction is or has been available at any step of product development.

Acquiring the relevant knowledge and expertise is commonly carried out by a knowledge engineer. Extracting the knowledge from the expert and representing it in a knowledge base using a proper conceptualization, however, cannot take place without first overcoming some obstacles.

2.1. Knowledge bottlenecks

Collecting and interpreting knowledge constitutes a major issue when building knowledge-based DSS. It is the often-referred bottleneck in the expert system development (Feigenbaum & McCorduck, 1983). There are several reasons why capturing knowledge is such a difficult undertaking.

2.1.1. Information bias

Knowledge that can be expressed in words and numbers only represents the tip of the iceberg of the entire body of possible knowledge (Polanyi, 1967). Therefore, the first and main obstacle to knowledge acquisition is the knowledge engineering paradox (Liebowitz, 1993). The knowledge and skills that constitute expertise in a particular domain are hidden in the head of experts in a tacit form. Tacit knowledge is the knowledge that people have acquired through experience over a long period of time, during which it has been processed by their subconscious. Usually, the more competent these experts become, the less able they are to describe the knowledge they use to solve problems.

Furthermore, tacit knowledge is informal and not-codified. Experts are able to make complex judgments rapidly by combining pieces of basic knowledge together in an unconscious process and without restating each step in the reasoning process. But when they attempt to explain their reasoning, it takes often an incomplete and non-sequential form that is not suitable for machine processing.

Another reason that can induce the knowledge bias lies in the way knowledge is collected. If it is done through observations, experts may change their behavior when they are being observed. If knowledge acquisition is done through interviews, information can be excluded simply because of the way the question is being asked. When a question is too specific, it does not allow all the information to be collected. Even when it does, knowledge engineers usually do not have the sufficient knowledge to fully comprehend and re-transcript the expert's answers.

2.1.2. Lack of willingness to share knowledge

Knowledge expands when people share their knowledge and experience with each other. Not distributing knowledge can induce heavy costs for companies in the form of production losses, damage to equipment, and reduction in product quality. Even accidents occurring in factories can often be attributed to a lack of worker's experience or inadequate training.

Sharing knowledge, however, does not come naturally to many employees as it is often said that knowledge is power and people can be reluctant to give up what makes them unique and inexpendable in an organization. Low morale, conflict, and mistrust are some of the barriers to people's willingness to share, but knowledge sharing can be encouraged by giving people credit for their ideas (Verkasalo, 1997). In times when many companies are reducing valuable workforce, the challenge is to create an atmosphere that fosters knowledge sharing and allows employees to leave a legacy that will eventually help the organization long after they leave.

Sometimes the unwillingness to share knowledge relates to company policies as these have ever stricter confidentiality rules. These rules, however, can also be the result of poor knowledge management as some companies are not able to separate sensitive from non-sensitive information.

2.1.3. Availability of knowledge

"The scarcest of corporate resources are less often the financial funds but rather the knowledge and expertise of the people in the front line" (Bartlett & Goshal, 1994). Experts are not always known, have only little time to spare, and since their knowledge is tacit, they might not even be aware of their expertise or how it can be valuable to others.

Today's global working conditions create an additional difficulty in reaching experts. Each expert does not know everything and the ones who do have the relevant knowledge might be located at the other end of the world or across the street at the subcontractor's plant. Surprisingly, the latter ones are not always the least difficult to reach. Distributed decision making becomes therefore a major issue (Verkasalo, 1997). Knowledge must be easily accessible because decisions made at one place have an impact on the work of other teams world-wide.

2.2. Knowledge extraction

Experts possess many different types of knowledge and numerous techniques have been developed to facilitate the extraction process. Tacit knowledge is considered more valuable because it incorporates a lot of additional learning related to individual behavior and perception. It is, however, difficult to separate it from its possessor and effective transfer of tacit knowledge generally requires extensive personal contact and trust. Therefore, face-to-face discussions remain the most widely used way of transferring knowledge as they have the ability to make tacit knowledge more explicit by allowing the expert to provide a context to his actions. But expert interviews and other manual techniques are not always possible and depend very much on the knowledge engineer's own understanding of the domain. The challenge in a global company is therefore to develop tools and methods that enable experts to be their own knowledge engineers or to provide multi-level interfaces adapted to the understanding of individual non-expert users.

Under these circumstances three topics have been studied to improve the efficiency of the knowledge extraction process. First, knowledge representation intends at providing the proper formalization of knowledge from all the available sources. Methodologies that provide guidelines for systematic and possibly automatic knowledge extraction activities are another one. Finally, the third development comes from an increasing use of software tools and more specifically user interfaces to aid the acquisition process.

2.2.1. Knowledge representation

For humans, learning is not simply a matter of acquiring a description. It involves taking something new and integrating it fully with existing thought processes. Thus, the ease of solving a problem is also determined by the way information is encoded into the memory. When building a DSS, the vocabulary used by experts is often inadequate for problem-solving because it is not understandable by others and even less by machines. Finding the right knowledge representation is therefore a vital task when generating a knowledge-base and an essential aspect of sharing and manipulating knowledge.

The basic model for knowledge engineering has long been that the knowledge engineer mediates between the expert and the knowledge-base, eliciting and encoding pieces of knowledge into an abstract form. To some extend, computer programs can also use forms of concept learning, which is the acquisition from examples of structural descriptions that can support different kinds of reasoning (MacDonald & Witten, 1989). There are, however, almost as many representations as there are systems. Nevertheless, a number of generic knowledge representations have been constructed, each having application across a number of domains (Holsapple, Tam, & Whinston, 1989). These generic representations are based on the idea that different problems can require similar tasks. For example, medical diagnosis and fault detection can both be categorized as generic classification tasks. The two problems, though different in their domain areas, are structurally similar in their use of diagnosis rules as seen in Fig. 1.

More generally, as the lack of understanding by non-experts in the knowledge elicitation process is an important issue in knowledge engineering, automatic elicitation of knowledge, if possible, offers great advantages in terms of knowledge database generation.

2.2.2. Automatic knowledge extraction

Automatic knowledge extraction methods make it possible to build a knowledge base with no need for a knowledge engineer and only very little need for an expert, mainly for validation purposes. Induction learning for example is a way to learn from experiments without having any predefined rules. The knowledge base is simply generated over time by generalization of the accumulated specific cases and workers can use knowledge of previous solving experience to clarify the causes and take appropriate action. When measurement data are available, the causes and possible solutions for a problem are usually also hidden in the databases. Knowledge discovery in databases (Klosgen & Zytkow, 2002), which is the automatic extraction of implicit and interesting patterns from large data collections can then be used to improve the knowledge-base.

Automatic knowledge extraction methods, however, face the knowledge acquisition dilemma: If the system is ignorant, it cannot raise good questions; if it is knowledgeable enough, it does not have to raise questions. Interview metasystems (Kawaguchi, Motoda, & Mizoguchi, 1991) can here provide a scalable approach to expert interviews by raising questions automatically to elicit new knowledge and refine the domain model. Furthermore, interviewing techniques can be extended through the use of graphical rather than numerical data entry to provide an interactive graphical elicitation environment within which the experts can distinguish cases (Gaines, 1993).

Because the domain knowledge is often very specific, knowledge acquisition is a labor-intensive task. In an attempt to standardize the knowledge acquisition process,

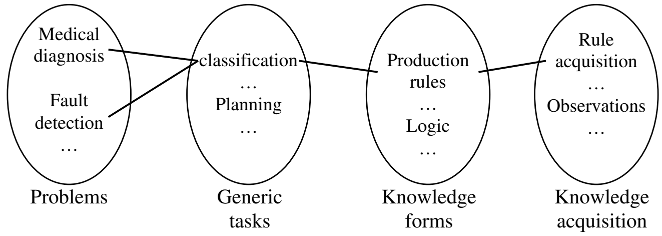

Fig. 1. Problem–task–representation–acquisition mapping (Holsapple et al., 1989).

generic acquisition shells have been developed (Chien & Ho, 1992) and extended with methods for updating incomplete or partially incorrect knowledge bases (Tecuci, 1992) (Winter, 1992) (Su, Zhang, Hou, & Pan, 2002). The work has also been facilitated by studies on the automatic acquisition of shallow knowledge, which is the experience acquired heuristically while solving problems (Okamura, Baba, Takahashi, & Shiomi, 1991). In some cases, instead of suppressing the knowledge engineer as an intermediary, the focus has been on compensating for his lack of domain knowledge, and the resulting knowledge base is accurate and complete (Fujihara, Simmons, Ellis, & Shannon, 1997).

2.2.3. User interface

To work efficiently with a system, the user needs to be able to control it, but also to assess its state so that he can define the proper course of action. In DSS users are often presented with an exhaustive amount of data upon which they have to make decision without necessarily having the proper understanding or knowledge to do so. In this context, the user interface (UI) can be seen as the dialogue component of a DSS that facilitates information exchange between the system and its users (Bálint, 1995).

The choice of an interface depends on many factors as interfaces are usually very context-specific tools. There are however only few reasons leading to its inadequacy (Norcio & Stanley, 1989). Mainly, the UI is often seen as the incidental part of the system. Consequently it is frequently not well suited to the system or to the user, and more often to neither. As interfaces are objects lying between domains of expertise, only team work involving specialists of different fields can place the user back in the center of the design process (Wills, 1994). Usability is the degree to which the design of a particular UI takes into account the psychology and physiology of the users, and makes the process of using the system effective, efficient and satisfying.

For its response to be understandable, a knowledge-based DSS should be able to tailor its response to the needs of the individual. Flexibility has therefore the potential to improve usability by taking into consideration the knowledge of the user, and also the knowledge of the interactions, the task/domain, and the system. UI adaptability can be achieved in a static or dynamic way (Jerrams-Smith, 1990). In static adaptability, the user can modify the interface if the behavior of the system is unsatisfactory. Although this may produce a better interface, it leaves the burden of adapting to the user. In the case of dynamic adaptability the interface works differently depending on the current context for example by mapping user's actions to what they intend to do (Eberts, 1991) or through concept mapping by matching what users want to what they need (Lind & Marshak, 1994). Adaptable UI are, however, no panacea as they can undermine the user's confidence in the information given to him. A solution to this problem investigated in the case study is to make the content adaptable instead of the interface itself in an attempt to adapt the embedded problem while keeping a fixed framework. This introduces the concept of explanatory power, which refers to the ability of a KBS to explain itself through content-based or interface-based enhancements (Nakatsu & Benbasat, 2003). For example, explanation may be presented to the user in different forms, but the transparency and flexibility of the reasoning will be increased by using graphical hierarchies instead of the equivalent flat interface to describe the structure of a rule base.

Different planning and design methodologies have been developed to insure that user specifications are taken into consideration (Wills, 1994) (Balasubramanian, Turoff, & Ullman, 1998) (McGraw, 1994). The following steps, also used in the case study, aim at building a separate methodology to develop user interfaces for knowledge acquisition:

- Identify and characterize the real users.
- Define a work process model.
- Definition of a general fault model.
- Design of a prototype.
- Testing, debugging, and redesigning.

3. Case study

3.1. Context

The company involved in this case study is an electronic subcontractor lacking the resources to engage into long-term process improvement and preferring therefore to focus efforts on end-of-line quality monitoring. Process control is left to the operators who make adjustments only based on their experience and personal knowledge of the production line. This means that the tuning of the system and consequently the quality of the product depend very much on human interpretation of machine problems. A DSS integrating this "know-how" could lower the variability inherent to human choices and greatly improve the efficiency of any response when a problem occurs.

The proposed system tries to provide the understanding and formalization of the parameters influencing the quality of the products, which are needed to improve the operative quality. The first step towards developing a control tool is to solve the defect traceability problem, not only in terms of data, but most of all in terms of knowledge from the production floor. This is achieved through an Expert Knowledge Acquisition System (EKCS) recording information about breakdowns each time a machine is stopped. Cross-analysis of the subsequent enriched information with measurement data can then greatly improve the ability to control the production as described in Fig. 2.

3.2. Implementation

The difficulty in modeling knowledge in a simple form makes its acquisition very challenging and requires standardized and rigorous methods. Eventually the aim is to integrate knowledge acquisition as the part of the expert's work in such a way that it becomes systematic without giving him the impression to be compulsory. This can be achieved by adding automated knowledge extraction features to daily used and user-friendly interfaces. Several design methodologies for these interfaces emphasize doing it right the first time, but it is our belief that an approach based on fast prototyping is better suited for creating tools that are both user and context specific. Furthermore, prototype versions have the advantage to allow the identification of problems that can be addressed when designing a more complete version. The design steps were already presented in an earlier paper (Gebus, Soulas, Fournier, & Ruusunen, 2004).

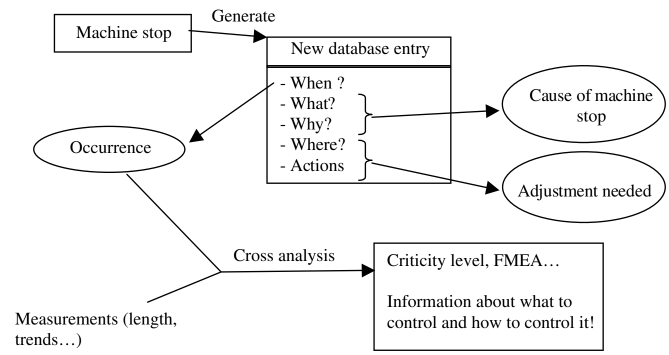

Fig. 2. Flowchart representing the expert knowledge collecting system.

EKCS was implemented and tested on the factory floor for a period of two weeks, during which all machine stops were systematically recorded. Analysis of that data and feedback from operators made it possible to identify four different categories of problems.

- Misunderstanding of the task is a common problem that arises when users are not provided enough training or when the system is not sufficiently self-explicative.
- Unexpected situations are the result of unforeseen problems.
- Expected unreferenced situations differ from the previous category in a way that they represent the gaps in the initial fault model.
- Lack of motivation can be seen from the rate at which the system has been used. 183 machine stops were recorded out of which only 70 were commented on.

3.2.1. Overall structure of the improved system

The DSS involves different kind of users, each of which have their own dedicated interface depending on the kind of access needed. In addition to the operator interface and the supervisor interface (aimed at the quality department), a specific interface has been created for updating the system. Fig. 3 describes the information flows between the different entities. As an information sharing system

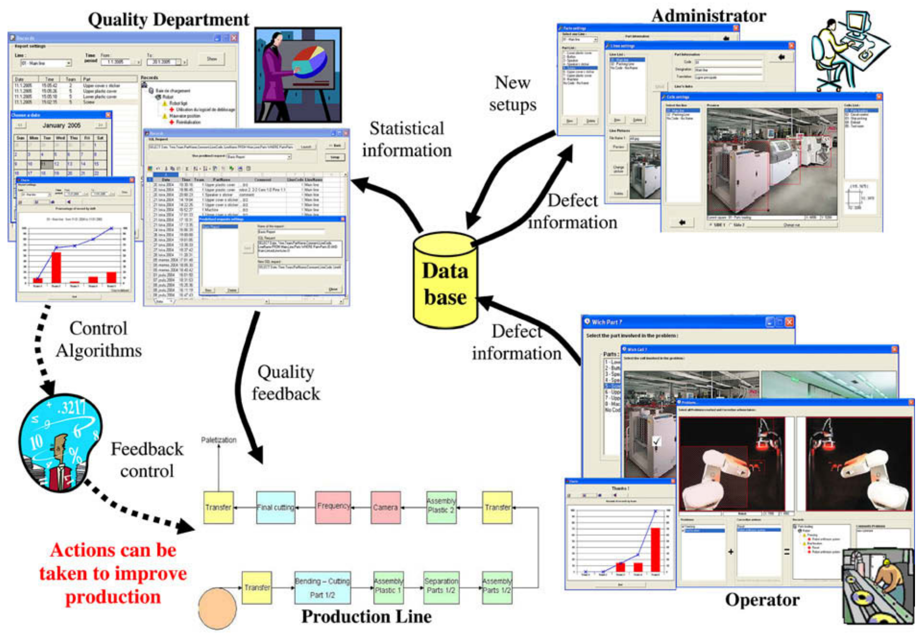

Fig. 3. General structure of the DSS with information flows between the different entities.

benefiting all the users, only closed loops of information flows should be present. Operators provide defect information that is stored in a database. The quality department can access any relevant historical data to produce statistical information. After analysis, quality feedback is generated and sent back to the production line. An administrator uses defect information casually for updating the system.

From the practical point of view, the three interfaces use a unique database allowing an automatic and immediate update of the system. This database is stored on a SQL server providing the needed flexibility that was missing in the prototype version. With such a structure, it is now possible to store not only information about date, time, defect and corrective actions, but also all the settings relative to the certain production line. This was added in order to make the administrator interface a fully integrated subsystem.

3.2.2. Some features in details

The administrator interface was designed so that new setups can be created easily and stored in the database to be used by the two other interfaces. Updating is done by selecting digital photographs of the production lines and cells. Since relevant knowledge concerns the localization and description of defects, this interface allows the administrator to create clickable areas that are then associated with the types of the defects and possible solutions.

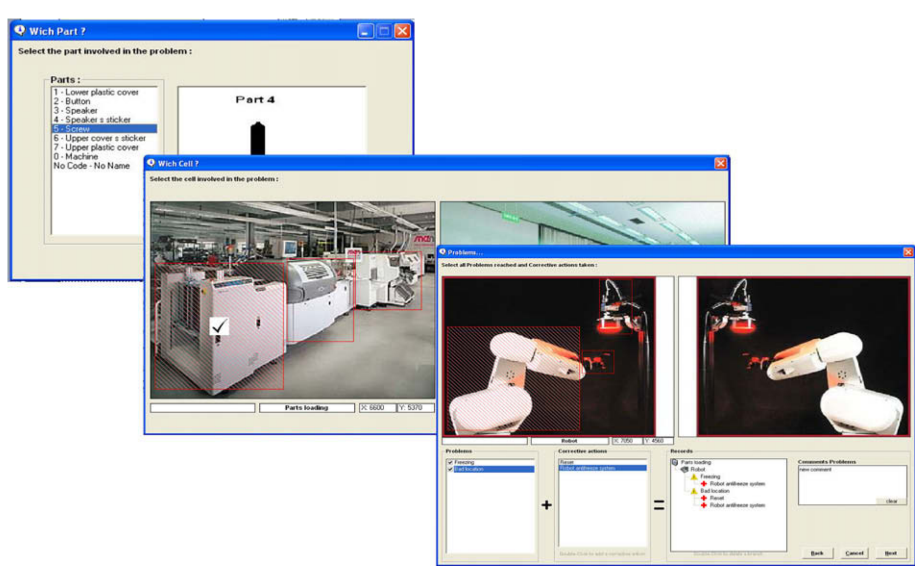

Fig. 4. Interface for fault selection.

Of the three interfaces, the one for the operators is the simplest and the most intuitive. Setup options are available but these are done only once when the system is implemented on a new production line or when pictures have been updated by the administrator, in which case they will be automatically downloaded to the local computer. In normal use, data input is done by choosing the product part from a list and zooming in on a defect by choosing a cell from pictures of the production line and a defect area from pictures of the cell as shown in Fig. 4. At this point, an interactive list of known causes and corrective actions is displayed as well as a comment window in the case of unreferenced problems. Selection of causes and actions generates automatically a decision tree shown in Fig. 5. Operators can access this information, along with the charts representing short-term quality and usage information.

The supervisor interface is more complex than the previous ones and offers three main options as shown in Fig. 6. The same kinds of reports as well as the same charts as in the operator interface are available, but without time restriction. It is therefore possible to visualize long-term trends or check some parts of historical data. In addition to these basic visualization properties, more advanced database exploration properties have been added. Customizable SQL requests can be created and run on the entire database before exporting the result to other software.

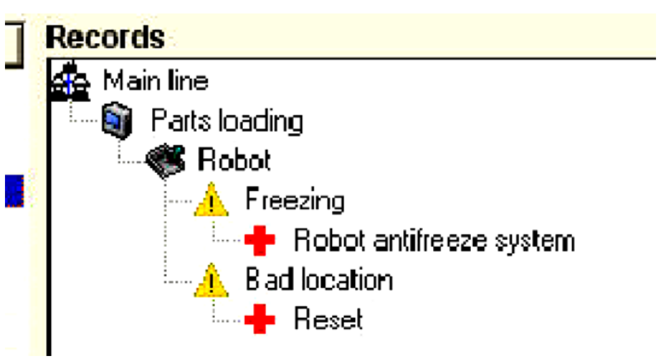

Fig. 5. Decision tree generated automatically.

3.2.3. Axes of improvement

When developing a DSS, users are the most unpredictable part involved in the design process. For that reason, problems encountered during the prototype phase are seldom concept-related problems, but are rather linked to the effective implementation of such a tool and its ability to communicate within a factory environment. From the feedback obtained during the prototype trial period, three different user-oriented axes of improvement were identified, Usability, Usefulness, and Usage (3U).

Usability improvements are mainly about making the information input simpler and faster in an environment that is really intuitive for users. Only this way it will become a part of the operator's everyday work instead of an additional burden. Furthermore, in an environment in constant evolution, design choices have to favor system portability. Checkboxes from the prototype version were therefore discarded as they are only suitable for very simple and most of all static cases. In more complex cases, they become confusing and affect the appearance of the interface. This goes against the idea that an adaptable interface should not evolve with the problem but rather remain static while presenting an evolving situation. The solution chosen to solve the portability problem was to locate defects by selecting clickable areas on a digital photograph of the production line. This in turn allows zooming in on a production cell and similarly to the list of problems and corrective actions. The use of digital photograph with clickable areas allows the needed flexibility and portability

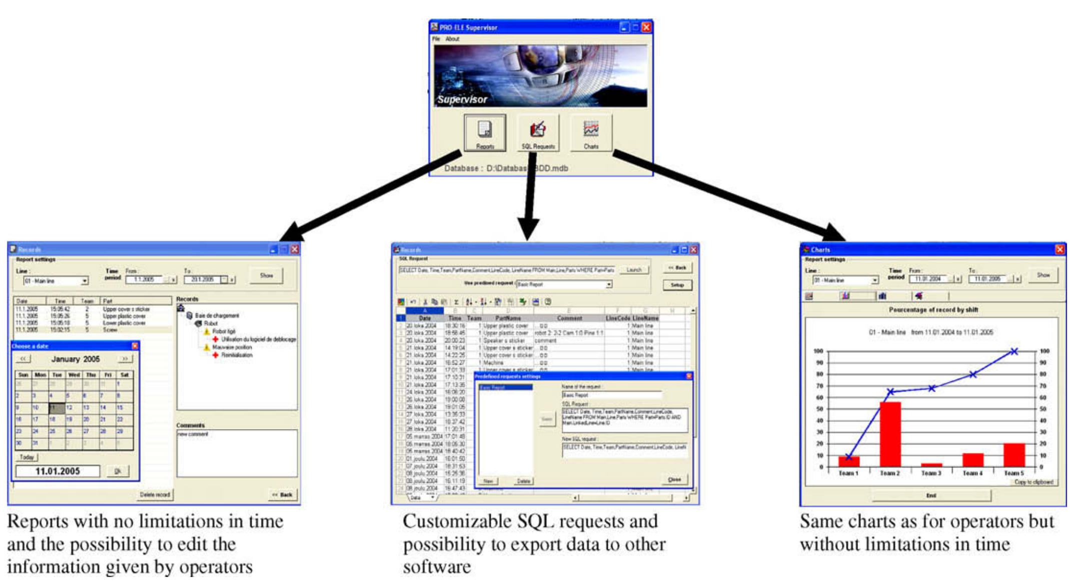

Fig. 6. Structure of the supervisor interface.

while keeping the environment and the interface very familiar, resembling a factory floor.

Usefulness is the very first and most important target of any system. This includes the primary users, but also all the secondary users, whose tasks are among others to maintain and update the system. Systems aimed for example at simplifying operators' work, but updated by design engineers, do not have any long-term continuity because the second group does not see any direct interest in performing the task. The problem of motivating every single user of the system into actually using it was achieved by transforming the DSS from a simple fault collecting system to a factory wide information sharing system that benefits all its users. This system can provide different levels of feedback depending on the expected added value from the current user. Short-term statistical feedback such as real-time failure rates are aimed at operators, whereas quality engineers can access long-term data through another dedicated interface.

Improved usage can partly be consequent to improved usefulness. In the case study, since defect information is sent back to the operators in various forms, it can be used during weekly meetings to discuss the production problems encountered. In the same way, quality engineers who now have a tool for automating some of their tasks and providing information tailored to their needs are more inclined to use it.

4. Results and discussion

Useful features for analyzing process conditions are often expressed in qualitative terms. Although it is not the most adequate form for computational processing, it is the form used by people during problem-solving activities. Collecting and interpreting this knowledge is recognized to be a very difficult task and many problems can arise during the knowledge acquisition process. Methods that have been developed to alleviate these problems are often extremely time-consuming, especially if knowledge extraction is done manually. Automatic extraction methods on the other hand are very context-specific and thus research often focused on rather narrow areas of the problem. From our knowledge about the topic, generalization of knowledge acquisition approaches in an attempt to create generic tools mostly resulted in tools that are poorly fitted for any specific case.

The case study presented in this paper is an attempt to solve some of the problems by transforming knowledge acquisition into a non-intrusive multi-level information sharing system that benefits directly to all its users. One concern is to provide it with the ability to communicate efficiently and transparently within a factory environment. A solution to simplify the information input could have been to fully automate the knowledge acquisition process through machine learning techniques. This, however, could have led to a system that eventually does not require anymore user involvement. Our approach proposes to focus on three user-oriented axes of improvement: *Usability*, *Usefulness*, and *Usage* (3U). These are more directions of improvements as it is very difficult in a context specific environment to provide precise general answers.

During the testing of the prototype version 183 machine stops occurred, out of which only 70 were commented on representing a usage rate of 38%. After having made the modifications, the final version of the system started to be used in a systematic way at different levels of the company, thus promoting employee involvement towards quality issues. Line operators, in particular, not only increased the usage rate to nearly 100%, but extend that use beyond the factory floor as defect reports generated automatically are now used during their weekly quality meetings. For them, the increased usage has been triggered by an improved usefulness.

The acceptation level for this new tool was also improved by the use of fully graphical interactions and digital pictures. As of our knowledge, the use of digital photography with clickable areas proposed as an extension to traditional graphical interfaces is a novel approach to knowledge acquisition for fault localization. This approach increases flexibility and portability of the system as it enables quick updates while keeping a familiar framework that resembles the factory floor. The user does not get confused with a changing interface as the adaptability of the content and the evolution of the situation take place within a static framework.

The new system enables the cross-analysis of data and expert information, which is a prerequisite for developing feedback control policies that will lead to a more efficient factory-wide knowledge and defect management. The supervisor interface, in particular, can be the backbone for the implementation of more complex monitoring algorithms as it has shown that any kind of previously stored data can not only be easily accessible, but also be processed by any chosen algorithm and the results can be sent back in various forms (charts, decision trees etc.). Such data analysis features are currently under development and will be the subject of a separate publication.

In the future, one can also imagine replacing manual feedback with control algorithms generating automatic feedback control. This is not possible in the current state of the system. Even if tasks performed by the quality department and data analysis were automated, proper actuators necessary to transform information into action on the production line would still be missing. Even if they did exist, automatic feedback control would still be highly dependent on line technology and therefore not portable.

5. Conclusion

An effective decision support system is essential to provide workers with information necessary to identify the causes of a problem and take appropriate action to solve it. This is achieved efficiently when integrating knowledge and more specifically shallow knowledge into the decision making process. In turn, integration of knowledge leads to decision support systems which will improve fault detection and recovery as well as defect-related communication within an organization.

When building such a knowledge-based system, knowledge acquisition is often considered to be the most time consuming part of the process. Defining the right interfaces for real-time automatic or semi-automatic knowledge acquisition can be a major problem. Eventually the approach that is used must be very human oriented as these interfaces have to be adapted to users with various degrees of knowledge. In addition to this, the complexity of any interface must be sufficient enough to catch the full scope of information, but simultaneously keep the data extraction process as simple as possible.

The general process for designing a knowledge acquisition interface applied to this case study presents the different tasks that have been undertaken and the problems encountered. Unlike traditional design techniques that emphasize doing it right the first time, the 3U approach proposed in this paper leads to a better match with user concerns. Knowledge acquisition software has been implemented on the production floor in a factory producing components for the electronics industry. Based on a test period, the knowledge gained from the use of this tool enabled defect classification and standardization. This is the first step towards cross analysis with monitored parameters from the production floor, leading eventually to online fault diagnosis.

References

Balasubramanian, V., Turoff, M., & Ullman, D. (1998). A systematic approach to support the idea generation phase of the user interface design process. In *Proceedings of the 31st Hawaii international conference on system sciences* (Vol 6, pp. 425–434).

Bartlett, C., & Goshal, S., 1994. Changing the role of top management: Beyond strategy to purpose. *Harvard Business Review,* Nov–Dec Issue.

Bálint, L. (1995). Adaptive interfaces for human-computer interaction: A colorful spectrum of present and future options. *IEEE International Conference on Systems, Man and Cybernetics, 1*, 292–297.

Chien, C. C., & Ho, C. S. (1992). Design of a generic knowledge acquisition shell. In *Proceedings of the IEEE international conference on tools with AI.* (pp. 346–349).

Dixon, N. M. (2000). *Common knowledge: How companies thrive by sharing what they know*. Harvard Business School Press.

Eberts, R. (1991). Knowledge acquisition using neural networks for intelligent interface design. *IEEE International Conference on Systems, Man, and Cybernetics, 2*, 1331–1335.

Feigenbaum, E. A., & McCorduck, P. (1983). *The fifth generation: Artificial intelligence and Japan's computer challenge to the world.* Addison-Wesley Longman.

Fujihara, H., Simmons, D. B., Ellis, N. C., & Shannon, R. E. (1997). Knowledge conceptualization tool. *IEEE Transactions on Knowledge and Data Engineering, 9*(2), 209–220.

Gaines, B. R. (1993). Eliciting knowledge and transferring it effectively to a knowledge-based system. *IEEE Transactions on Knowledge and Data Engineering, 5*(1), 4–14.

Gebus, S., & Juuso, E. K. (2002). Industrial utilization of linguistic equations for defect detection on printed circuit boards. In *Proceedings of the IECON'02 IEEE conference* (pp. 1887–1892).

Gebus, S., Soulas, A., Fournier, G., & Ruusunen, M. (2004). Design of an interface for knowledge acquisition on the production floor. In *Proceedings of the Eunite 2004 symposium* (pp. 139–144).

Gebus, S. (2006). *Knowledge-based decision support systems for production optimization and quality improvement in the electronics industry*. Oulu University press.

Holsapple, C. W., Tam, K. Y., & Whinston A. B. (1989). Building knowledge acquisition systems – A conceptual framework. In *Proceedings of the 22nd Hawaii international conference on system sciences* (Vol. 3, pp. 200–210).

Jerrams-Smith, J. (1990). User knowledge modelling for improved communication with KBSs. *IEEE Colloquium on AI in the User Interface*, 5/1–5/3.

Kawaguchi, A., Motoda, H., & Mizoguchi, H. R. (1991). Interview-based knowledge acquisition using dynamic analysis. *IEEE Expert*, 47–60.

Klosgen, W., & Zytkow, J. (2002). *Handbook of data mining and knowledge discovery*. New York: Oxford University Press.

Liebowitz, J. (1993). Educating knowledge engineers on knowledge acquisition. *IEEE International Conference on Developing and Managing Intelligent System Projects, 3*, 110–117.

Lind, S., & Marshak, W. (1994). Cognitive engineering computer interfaces: Part 1 – Knowledge acquisition in the design process. In *Proceedings of the NAECon 1994 Conference* (Vol. 2, pp. 753–755).

MacDonald, B. A., & Witten, I. H. (1989). A framework for knowledge acquisition through techniques of concept learning. *IEEE Transactions on Systems, Man, and Cybernetics, 19*(3), 499–512.

McGraw, K. L. (1994). Knowledge acquisition and interface design. *Software IEEE, 11*(6), 90–92.

Nakatsu, R. T., & Benbasat, I. (2003). Improving the explanatory power of knowledge-based systems: An investigation of content and interface-based enhancements. *IEEE Transactions on Systems, Man, and Cybernetics, 33*(3), 344–357.

Norcio, A. F., & Stanley, J. (1989). Adaptive human–computer interfaces: A literature survey and perspective. *IEEE Transactions on Systems, Man, and cybernetics, 19*(2), 399–408.

Okamura, I., Baba, K., Takahashi, T., & Shiomi, H. (1991). Development of automatic knowledge-acquisition expert system. In *Proceedings of the international conference on industrial electronics, control and instrumentation* (Vol. 1, pp. 37–41).

Polanyi, M. (1967). *The tacit dimension*. New York: Anchor Books.

Seabra Lopes, L., & Camarinha-Matos, L. M. (1995). A machine learning approach to error detection and recovery in assembly. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (Vol. 3, pp. 197–203).

Spanos, C. J., & Chen, R. L. (1997). Using qualitative observations for process tuning and control. *IEEE Transactions on Semiconductor Manufacturing, 10*(2), 307–316.

Su, L. M., Zhang, H., Hou, C. Z. & Pan, X. Q. (2002). Research on an improved genetic algorithm based knowledge acquisition. In *Proceedings of the 1st international conference on machine learning and cybernetics* (pp. 455–458).

Tecuci, G. D. (1992). Automating knowledge acquisition as extending, updating, and improving a knowledge base. *IEEE Transactions on Systems, Man, and cybernetics, 22*(6), 1444–1460.

Turban, E. (1995). *Decision support and expert systems: Management support systems*. Englewood Cliffs, NJ: Prentice-Hall.

Verkasalo, M. (1997). *On the efficient distribution of expert knowledge in a business environment*. Oulu university press.

Wills, C. E. (1994). User interface design for the engineer. In *Proceedings of the Electro/94 international conference*.

Winter, G. B. (1992). An automated knowledge acquisition system for model-based diagnostics. *Autotest Conf. Record.*, 167–174.

Özbayrak, M., & Bell, R. (2003). A Knowledge-based decision support system for the management of parts and tools in FMS. *Decision Support Systems, 35*(4), 487–515.