DEFECT-RELATED KNOWLEDGE ACQUISITION FOR DECISION SUPPORT SYSTEMS IN ELECTRONICS ASSEMBLY

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

Keywords: Decision support system, knowledge acquisition, quality, optimization, traceability, feedback.

Abstract: Real-time process control and production optimization are extremely challenging areas. Traditional approaches often lack in robustness or reliability when dealing with incomplete, inaccurate, or simply irrelevant data. This is a major problem when building decision support systems especially in electronics manufacturing, where blind feature extraction and data mining methods on large databases are common. Performance of these methods can 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 acquisition of shallow knowledge concerning everyday human interventions on the production lines, as well as on the conceptualization and factory wide sharing of the resulting defect information. Software with dedicated interfaces has been developed for that purpose. 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.

1 INTRODUCTION

A decision support system (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). Data, however, does not exist naturally in a factory; it has to be collected, stored, prepared, and eventually mined. Morover, it might be incomplete, inaccurate, or simply irrelevant to the problem that is being investigated thus leading to the inability of decision makers to efficiently diagnose many malfunctions, which arise at machine, cell, and entire system levels during manufacturing operations (Özbayrack & Bell, 2002). These difficulties might be overcome by taking into consideration knowledge about the environment, the task, and the user (Gebus, 2006).

Knowledge-based approach takes advantage of the fact that it is the people operating the process who are most likely to have the best ideas for its improvement. It is through the integration of these ideas into the problem solving approach that a solution for long term process improvement can be found (Seabra Lopes & Caraminha-matos, 1995). Additionally, as the use of knowledge and more generally qualitative information better explains the relationships between input process settings and output response, it is well indicated for improving the understanding and usability of DSS (Spanos & Chen, 1997). In this paper, we shall examine the possibility to integrate knowledge in general, and especially shallow knowledge, into the decision making process. Section 2 presents the problematic related to knowledge acquisition and knowledge-related improvements in man/machine interractions. Section 3 presents our contribution to that field through a case study that is followed by a discussion on the results and the conclusion.

2 KNOWLEDGE ACQUISITION

Unlike data, knowledge does exist naturally in the factory, but collecting and interpreting it constitutes a major issue when building knowledge-based DSS.

Gebus S. and Leiviskä K. (2007).
DEFECT-RELATED KNOWLEDGE ACQUISITION FOR DECISION SUPPORT SYSTEMS IN ELECTRONICS ASSEMBLY.
In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics*, pages 270-275

These tasks commonly carried out by a knowledge engineer are often referred to as the bottleneck in the expert system development (Feigenbaum & McCorduck, 1983).

First and main obstacle is the knowledge engineering paradox (Liebowitz, 1993). Knowledge and skills that constitute expertise in a particular domain is tacit. Furthermore, the more competent experts become, the less able they are to describe how they solve problems. Another contribution to the bottleneck is the lack of willingness to share knowledge. It is often said that knowledge is power and people can be reluctant to give up what makes them inexpendable (Verkasalo, 1995). Finally, knowledge availability constitutes another obstacle as experts are not always known and have little time to spare. Additionally, today's global working conditions make it hard reaching experts located at the other end of the world or across the street at the subcontractor's plant. Distributed decision making becomes therefore a major issue (Verkasalo, 1995).

Currently, face to face discussions are still 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. Three topics are commented upon here: knowledge representation, automatic knowledge extraction and the user interface.

2.1 Knowledge Representation

Experts reasoning is often incomplete and not suitable for machine processing. Creating the proper ontology is therefore an essential aspect of sharing and manipulating knowledge. Based on the notion that different problems can require similar tasks, a number of generic knowledge representations have been constructed, each having application across a number of domains (Holsapple et al., 1989). Common classes of knowledge representations are logic, semantic network, or production rules.

Computer programs can use forms of concept learning to extract from exemples structural descriptions that can support different kinds of reasoning (MacDonald & Witten, 1989). More generally, automatic elicitation of knowledge, if possible, offers great advantages in terms of knowledge database generation.

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, for example by using case-based reasoning. This poses however a 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. Scalable acquisition techniques such as interview metasystems (Kawaguchi et al., 1991) or interviewing techniques using graphical data entry (Gaines, 1993) can help overcoming this difficulty.

Because the domain knowledge is often very specific, knowledge acquisition is a labor-intensive task. For that reason, generic acquisition shells have been developed (Chien & Ho, 1992) and extended with methods for updating incomplete or partially incorrect knowledge bases (Tecuci, 1992) (Su et al., 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 et al., 1991), or by compensating for the knowledge engineer's lack of domain knowledge, so that the resulting knowledge base is accurate and complete (Fujihara et al., 1997).

2.3 User Interface

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. The user interface (UI) is 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 factor, but there are only few reasons for its inadequacy (Norcio & Stanley, 1989). Mainly, the UI is often seen as the incidental part of the system. Consequently it is not well suited to the system or to the user, and more often to neither. Usability can be seen as 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 DSS should be able to tailor its response to the needs of the individual. UI adaptability can be achieved by mapping user's actions to what they intend to do (Eberts, 1991) or need (Lind et al., 1994). This can undermine however the user's confidence in the information given to him. Adaptability can therefore be applied to the content instead of the interface itself by increasing the ability of a DSS to explain itself for example by using graphical hierarchies instead of the equivalent flat interface to describe the structure of a rule base (Nakatsu & Benbasat, 2003).

Different planning and design methodologies have been developed to insure that user specifications are taken into consideration (Wills, 1994) (Balasubramian et al., 1998) (McGraw, 1994). The following steps, also used in the case study, aim to build 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, and
- Test, debugging, and redesign.

3 CASE STUDY

The electronic subcontractor involved in this case study is lacking the resources for long-term process improvement. Process control is left to the operators who make adjustments only based on experience and personal knowledge of the production line. Thus tuning of the system and consequent 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 DSS tries to provide understanding and formalization of the parameters influencing the quality of the products, which are needed to improve the operative quality. Traceability in terms of know-how from the production floor is achieved through an Expert Knowledge Acquisition System (EKCS) recording the main breakdown information. Cross-analysis of the subsequent enriched information with measurement data can then improve the ability to control the production as described in Figure 1.

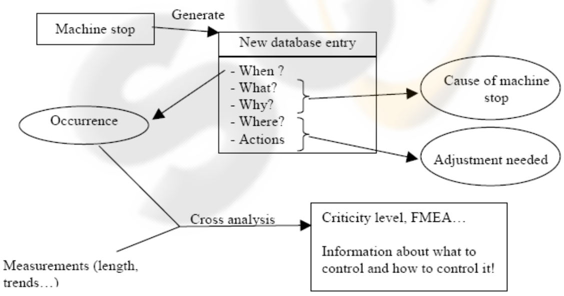


Figure 1: Flowchart representing the Expert Knowledge Acquisition System.

3.1 Prototype Version

The approach based on fast prototyping has already been described (Gebus, 2006). It is well suited for creating tools that are both user and context specific. This section serves as a reminder of the main points and conclusions concerning the prototype version.

Line operators are the main users and knowledge providers of the DSS, whereas quality engineers are interested in getting knowledge in a way, which is most suitable for a comparative analysis with measurement data. The process model describes the interactions between the process and the knowledge sources. The fault model analyzes the cause and effect relationships in the process model. Finally, a prototype interface is designed to guide the operator through the knowledge acquisition process using pictures of the product and check boxes representing the different cells of the production line. Fault information is entered manually.

Prototype versions are time consuming but have the advantage to allow the identification of problems that the knowledge engineer would otherwise be unaware of. All these problems can then be addressed when designing a more complete version.

3.2 Final Version

3.2.1 Axes of Improvement

Based on the feedback from the test period, three different user-oriented axes of improvement were identified, Usability, Usefulness, and Usage (3U).

Usability concerns mainly knowledge input that needs to be simpler, faster, and more intuitive. The input method also has to favor system portability as equipment is in constant evolution. In this context, checkboxes that affect the appearance of the UI are confusing and are replaced by clickable areas on digital photographs of the production line and cells. A defect is located by zooming in on an area, which is linked to a list of problems and corrective actions. This approach enables the needed flexibility and portability while keeping the environment and the interface very familiar, resembling a factory floor.

Usefulness probably is the most important target of any system. However, experience has shown that a system aimed at simplifying operators' work, but updated by design engineers, does not have any long-term continuity. Motivating every single user of the system into actually using is achieved by transforming the DSS from a simple fault collecting system to a factory wide information sharing system providing user-specific levels of added value.

Improved usage is consequent to improved usefulness. Defect information is sent back to the operators in forms that can be used during meetings to discuss encountered production problems. In the same way, quality engineers are more inclined to use a system that automates some of their tasks and provides information tailored to their needs it.

3.2.2 Overall Structure

In addition to specific interfaces for operators and quality engineers, a dedicated UI has been created for updating the system. As an information sharing system benefiting all the users, Figure 2 shows only closed loops of information flows. Operators provide defect information 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 only 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. Such a structure enables storing not only information about date, time, defect and corrective actions, but also all the settings relative to a certain production line. This was added in order to make the administrator interface a fully integrated subsystem.

3.2.3 Some Features in Details

The administrator interface was designed so that new setups can be created easily. Updating is done by selecting digital photographs of the production lines and cells, and creating clickable areas that are then linked to defect types and possible solutions.

Concerning the operator interface, emphasis has been put on simplicity and intuitiveness. Setup options are limited to choosing the database location and selecting the current line. This is done only once when the DSS is implemented on a new production line or when the digital photographs are updated. In normal use, data input is done by choosing the product part from a list, and choosing the defect area by zooming in on the pictures as in Figure 3. The list of known causes and corrective actions is made available as well as a comment window in case of unreferenced problems. Selection of causes and actions generates automatically a decision tree and updates charts representing short-term quality and usage information.

The supervisor interface is more complex and offers three main options as shown in Figure 4. Similar reporting capability as in the operator interface is available, but with no limitations in time. Long-term trends can be visualized from historical data. Furthermore, 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.

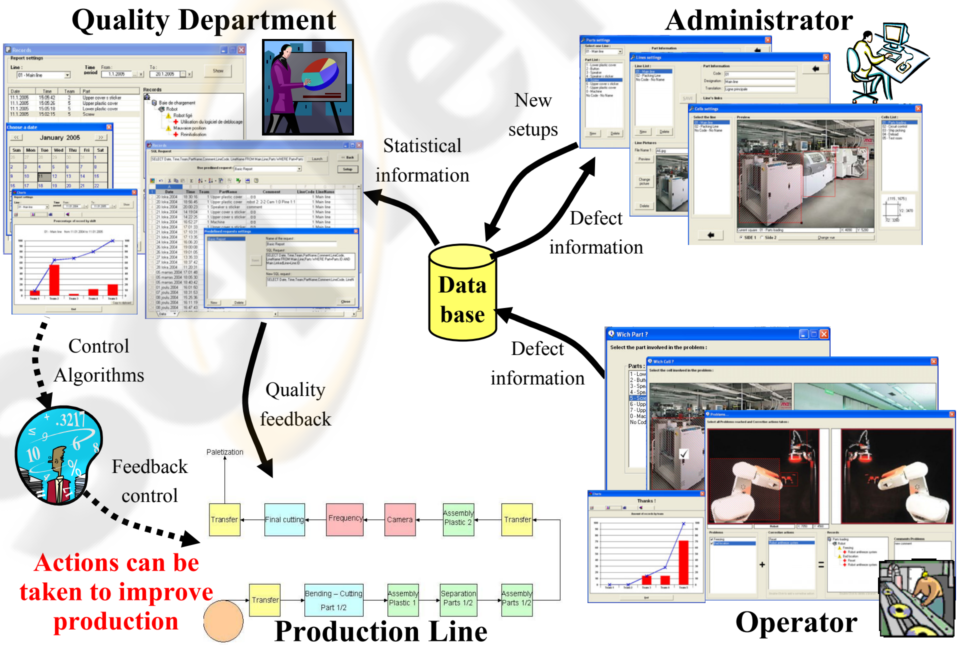


Figure 2: General structure of the DSS with information flows between the different entities.

4 RESULTS AND DISCUSSION

During the testing of the prototype version 183 machine stops occurred, out of which only 70 were commented on representing therefore a usage rate of 38%. After having made the modifications, the final version 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 to their weekly quality meetings. For them the increased usage has been triggered by an improved usefulness.

The acceptation level for this new tool is based on fully graphical interactions and digital pictures. These enable quick updates of the system while keeping a familiar framework. It also shows that in order to satisfy the user needs, the real challenge for an adaptable interface is not to evolve with the problem, but rather to remain static while presenting an evolving situation.

The new systems 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 implementing monitoring and control algorithms. 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.).

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 as proper actuators necessary to transform information into action on the production line are missing. Even if they did exist, automatic feedback control would still be highly dependent on line technology and therefore not portable.

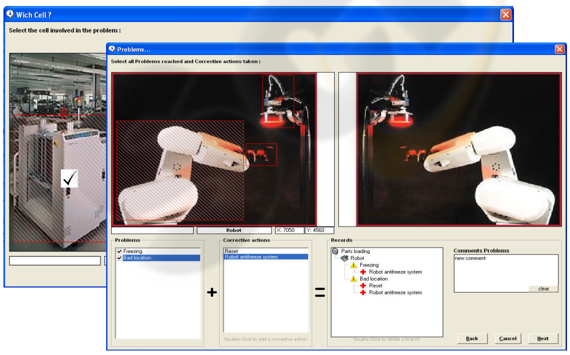


Figure 3: Interface for fault selection.

5 CONCLUSION

When building a knowledge-based system, the approach that is used has to be very human oriented. Defining the right interfaces for real-time knowledge acquisition can be a major problem. They 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 section 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 on-line 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. *31st Hawaii Int. Conf. on System Sciences*. vol. 6, pp. 425 – 434.

Bálint, L., 1995, Adaptive Interfaces for Human-Computer Interaction: A Colorful Spectrum of Present and Future Options. *IEEE Int. Conf. on Systems, Man and Cybernetics*. vol. 1, pp. 292 – 297.

Chien, C.C., Ho, C.S., 1992, Design of a Generic Knowledge Acquisition Shell. *IEEE Int. Conf. on Tools with AI*. pp. 346 – 349.

Eberts, R., 1991, Knowledge Acquisition Using Neural Networks for Intelligent Interface Design. *IEEE Int. Conf. on Systems, Man, and Cybernetics*. vol.2, pp. 1331 – 1335.

Feigenbaum, E.A., McCorduck, P., 1983, *The fifth generation: artificial intelligence and Japan's computer challenge to the world.* Addison-Wesley Longman Publishing Co.

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., 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. *22nd Hawaii Int. Conf. on System Sciences*. vol. 3, pp. 200 – 210.

Kawaguchi, A., Motoda, H., Mizoguchi, H.R., 1991, Interview-Based Knowledge Acquisition Using Dynamic Analysis. *IEEE Expert*: 47 – 60.

Liebowitz, J., 1993, Educating Knowledge Engineers on Knowledge Acquisition. *IEEE Int. Conf. on Developing and Managing Intelligent System Projects*. pp. 110 – 117.

Lind, S., Marshak, W., 1994, Cognitive Engineering Computer Interfaces: Part 1 – Knowledge Acquisition in the Design Process. *NAECon 1994 Conf.* 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. *Int. Conf. on Industrial Electronics, Control and Instrumentation*. vol.1, pp. 37 – 41.

Seabra Lopes, L., Camarinha-Matos, L.M., 1995, A Machine Learning Approach to Error Detection and Recovery in Assembly. *IEEE/RSJ Int. Conf. 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. *1st Int. Conf. on Machine Learning and Cybernetics*. pp. 455 – 458.

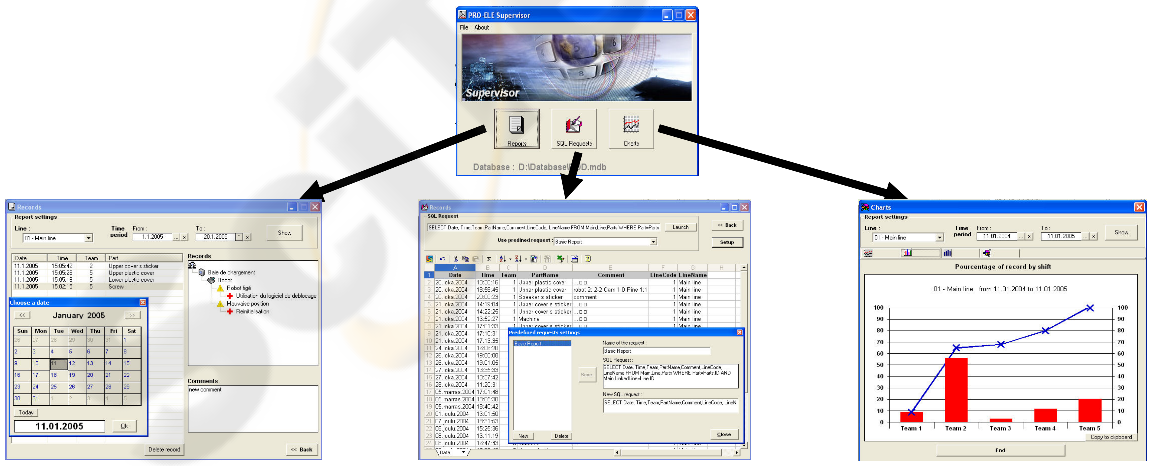
Tecuci, G.D., 1992, Automating Knowledge Acquisition as Extending, Updating, and Improving a Knowledge Base. *IEEE Transactions on Systems, Man, and cybernetics* 22(6): 1444 – 1460.

Turban, E., 1995, *Decision support and expert systems: management support systems*. Englewood Cliffs, N.J., 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. *Electro/94 International Conference*.

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.



Reports with no limitations in time and the possibility to edit the information given by operators

Customizable SQL requests and possibility to export data to other software

Same charts as for operators but without limitations in time

Figure 4: Structure of the supervisor interface.