OULU 2006
C 255

ACTA UNIVERSITATIS OULUENSIS

Sébastien Gebus

KNOWLEDGE-BASED DECISION SUPPORT SYSTEMS FOR PRODUCTION OPTIMIZATION AND QUALITY IMPROVEMENT IN THE ELECTRONICS INDUSTRY

FACULTY OF TECHNOLOGY,
DEPARTMENT OF PROCESS AND ENVIRONMENTAL ENGINEERING,
CONTROL ENGINEERING LABORATORY,
UNIVERSITY OF OULU

ACTA UNIVERSITATIS OULUENSIS
C Technica 255

SÉBASTIEN GEBUS

KNOWLEDGE-BASED DECISION SUPPORT SYSTEMS FOR PRODUCTION OPTIMIZATION AND QUALITY IMPROVEMENT IN THE ELECTRONICS INDUSTRY

Academic dissertation to be presented, with the assent of the Faculty of Technology of the University of Oulu, for public defence in Kuusamonsali (Auditorium YB210), Linnanmaa, on September 22nd, 2006, at 12 noon

OULUN YLIOPISTO, OULU 2006

Acta Univ. Oul. C 255, 2006

Supervised by
Professor Kauko Leiviskä

Reviewed by
Professor Tapio Frantti
Professor Kari Koskinen

ISBN 951-42-8204-3 (Paperback)
ISBN 951-42-8205-1 (PDF) http://herkules.oulu.fi/isbn9514282051/
ISSN 0355-3213 (Printed)
ISSN 1796-2226 (Online) http://herkules.oulu.fi/issn03553213/

Cover design
Raimo Ahonen

OULU UNIVERSITY PRESS
OULU 2006

Gebus, Sébastien, Knowledge-based decision support systems for production optimization and quality improvement in the electronics industry
Faculty of Technology, University of Oulu, P.O.Box 4000, FI-90014 University of Oulu, Finland, Department of Process and Environmental Engineering, Control Engineering Laboratory, University of Oulu, P.O.Box 4300, FI-90014 University of Oulu, Finland
Acta Univ. Oul. C 255, 2006
Oulu, Finland

Abstract

For the past few years, electronics manufacturing may have been the victim of its own success. Unlike in heavier industries, rationalization is a concept that was unknown in the sector until only a few years ago and even now, many companies are struggling with cost-cutting measures. Production systems in electronics manufacturing need to be highly flexible because of a varying and evolving environment. Therefore real-time process control and, possibly as a result, production optimization are extremely challenging areas. Traditional approaches often do not work due to a lack of robustness or reliability.

For this reason, a new generation of decision support systems is needed in response to some specific problems. The thesis addresses topics such as design of intelligent interfaces for knowledge acquisition and elicitation, use of that knowledge for improved data analysis and diagnostics, real-time feedback control, self-tuning capabilities, and evaluation of optimization methods in discrete processes. Topics covered therefore include the whole scope of a decision support system, from its design through to the evaluation of its performance as well as interaction capabilities as a vehicle for sharing information.

The aim of this research is to streamline the development of a new generation of decision support systems by providing tools and methods for a better integration of knowledge in an evolving environment. The main interest lies not only in improved data analysis, but also in better formalization and use of diagnosis. Case studies presented in this thesis demonstrate the practical feasibility of such an approach.

Keywords: DSS, knowledge-based decision support system, process optimization, quality improvement

To my family, my friends
and everybody who supported my work

Acknowledgements

The content of this thesis covers a five-year period between the years 1999 and 2006 at the Control Engineering Laboratory, University of Oulu. I am indebted to Professor Kauko Leiviskä for creating the opportunity to carry out this work. His guidance and more specifically the time he accorded to my work, especially over the past year, have been invaluable.

I also wish to thank Professor Tapio Frantti and Professor Kari Koskinen for reviewing the thesis.

Mika Ruusunen is acknowledged for many useful discussions related to both my work, which made him co-author of most of my articles, and to the everyday problems that anybody who is living for such a long time in a foreign country inevitably has to face. I am also grateful to Esko Juuso who supervised my research in my first year in Finland and was my primary source of information for everything concerning Linguistic Equations.

Since the research was very much oriented toward industrial applications, I am thankful to the different companies involved in the various projects in which I took part over those years. More specifically, I wish to thank Dr Matti Verkasalo and Vesa Similä, respectively my supervisor and my manager during my engineering traineeship in 1999 on my first stay in Finland. Their attitude toward my work and belief in my ideas were the main motivation for me to start a thesis.

My first stay in Finland would not have been possible without the staff of the French Institute for Advanced Mechanics (IFMA), and in particular Professor Henri Pierreval. They gave me the opportunity to spend one year abroad. This engineering school, from which I graduated in 2000, sends almost all its students abroad on one-year exchange programs. Over the years, I therefore also had the opportunity to supervise the work of those following in my footsteps. Naming all of them would take too long, but I am thankful for all their work related to the practical implementation of the methods presented in this thesis.

Finally, I would like to thank Gaëlle Brideau for her support and understanding. Without her, I would not have been able to keep up for so long the 12 to 16 hours a day plus weekends that I spent writing the thesis.

Oulu, September 2006 Sébastien Gebus

Abbreviations

3U	Usability, Usefulness, Usage
AI	Artificial Intelligence
ANN	Artificial Neural Network
AOI	Automated Optical Inspection
ARL	Average Run Length
AXI	Automated X-ray Inspection
BIST	Built-In Self Test
CBR	Case-Based Reasoning
CUSUM	Cumulative Sum
DHTML	Dynamic HTML
DSS	Decision Support System
EKCS	Expert Knowledge Collecting System
EWMA	Exponentially Weighted Moving Average
FME(C)A	Fault Mode and Effect (and Criticality) Analysis
GUI	Graphical User Interface
HCI	Human Computer Interface/Interaction, part of HMI
HMI	Human Machine Interface/Interaction
HTML	Hyper Text Mark-up Language
ICT	In-Circuit Test
IDEF	Integrated DEFinition language
KBS	Knowledge-Based System
LAN	Local Area Network
LE	Linguistic Equation
MDA	Manufacturing Defect Analyzer
MMS	Man-Machine System
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
MVI	Manual Visual Inspection
PCA	Principal Component Analysis
PCB	Printed Circuit Board
PDCA	Plan-Do-Check-Act

PLS	Partial Least Squares
ppm	Part Per Million
R2R / RbR	Run-to-Run / Run-by-Run
SADT	Structured Analysis and Design Technique
SPC	Statistical Process Control
SQC	Statistic Quality Control
SQL	Short Query Language
TQM	Total Quality Management
UI	User Interface
σ	sigma or also the standard deviation

List of symbols

Chapter 4

X, Y	Variables
a, b	Interval limits
x_i	Data point
X_i	linguistic value associated with x_i
a_i, b_i	Coefficients of a second order polynomial
c	Mean or median value of a data set
A	Interaction matrix
B	Bias vector
σ	standard deviation
k	Corrective factor

Chapter 5

Dx_avg	Average sample shift in X direction
X_avg	Standard value of the average shift
X_max	Maximum value of the standard shift
X_max_refno	Reference number of the component causing X_max
X_min	Minimum value of the standard shift
X_min_refno	Reference number of the component causing X_min
k	Corrective factor used to take into account the sample size
a1, a2	Weighting coefficients
T1, T2	Warning and control limits
N	Number of history data available (sample size)
I_1, I_2, I_3, I_4, I_5	Quality indices

Contents

1 Introduction

Recent evolutions of business models in the electronics industry suggest that electronics manufacturing may have been the victim of its own success. Increasing globalization of the economy is imposing tough challenges on manufacturing companies in general and the electronics sector in particular. Real-time process control and, possibly as a result, production optimization are extremely challenging areas. Traditional approaches often do not work due to a lack of robustness or reliability. A new generation of decision support systems is therefore needed in response to some specific problems.

1.1 Research problem

Production systems in electronics manufacturing need to be highly flexible because of a varying and evolving environment. This ability to produce highly customized products, in order to satisfy market niches, requires the introduction of new features in automation systems. Flexible manufacturing systems must be able to handle unforeseen events. The complexity of flexible manufacturing processes makes supervision and maintenance tasks difficult to perform by human operators.

Much work has been done to assist human operators in supervising processes and to improve both process and product quality. Problems in doing so are manifold as this kind of tool should also encompass human knowledge as well as advanced communication capabilities.

In this thesis the research problem is formulated as follows:

> *How to acquire knowledge that is relevant to quality improvement efficiently and integrate it into the decision making process at an operational and organizational level? The thesis will focus on three aspects of this problem: acquisition and sharing of defect-related knowledge, utilization of knowledge for fault diagnostic purposes, and generation of information for short-term feedback control.*

1.2 Research assumptions

Knowledge exists in any given organization and can take various forms. When this knowledge belongs to a physical person, it is also referred to as expertise or experience. It is what makes that person efficient in doing his or her assigned task. Sometimes knowledge resides also within non-human objects either in a straight and formal way (*e.g.* expert systems), or a conceptual way. Design information, for example, can be seen as an expression of knowledgeable choices made by the designer. In other words, we could say that knowledge exists as soon as human interaction is or has been available at any step of product development.

In the electronics industry, the rapidly evolving environment and short life cycle of products do not allow the extensive build-up of operational experience on the factory floor. Under these circumstances the proper formalization of the knowledge from all the available sources is a vital task. As a result, quality-related decisions rely heavily on measurement data that focuses often on product rather than on process characteristics. Decision support systems that make the best out of various forms of sometimes imprecise knowledge are needed to enhance the outcome of data analysis.

The research assumptions can be summarized as follows:

In any manufacturing organization, human knowledge, whether it is generated at an operational level on a factory floor or encompassed within higher-level conceptualization forms, exists and can be used to describe important qualitative and quantitative aspects of the process. In the electronics industry, the potential quality improvement deriving from untapped knowledge requires a new form of decision support systems based on data and knowledge.

1.3 Hypothesis

Process optimization and, as a result, quality improvement are tasks that rely heavily on factory floor human expertise. Yet this expertise, along with other forms of knowledge, remains largely unused in the electronics industry where automated quality monitoring is concerned. We consider that the relevant knowledge can be extracted in an automatic or semiautomatic way and used to improve the outcomes of more conventional data based approaches. Especially in a data-poor environment, expert knowledge can improve diagnostic and maintenance abilities from the repair and rework point of view by allowing detection and physical localization of faults. This is a clear advantage over existing end of line testing strategies. It can also provide faster feedback as there is no need to collect extensive historical data. Finally, as data analysis does not provide a natural language for human beings, we consider that integrating knowledge within the structure of a decision support system will enhance its ability to communicate and increase the involvement of all the actors on quality all along the production process, from the designer to the quality engineer. In summary, the research hypothesis can be formulated as follows:

Better integration of knowledge leads to decision support systems which will improve fault detection and recovery as well as defect-related communication within an organization.

1.4 Scope of applicability

The aim of this research has been to develop reliable tools and methods to be integrated in decision support systems in the specific context of electronics manufacturing. Discrete processes, rapidly evolving environments, abundant data but usually only partial knowledge scattered all along the production process are among the main characteristics. Development and implementation of the tools presented in this thesis were made using programming languages such as Visual Basic, DHTML, and Delphi, as these allow the creation of simple interfaces. They also enable easy and fast communication with existing systems and databases and are therefore suitable for real-time applications. Apart from section 3 which concerns knowledge acquisition in the form of fault characterization and tracking in a general sense, the rest of this thesis focuses on defects related to electronics assembly lines, either before soldering (section 5) or at the final testing stage (section 4).

1.5 Background to the research and results

The research activities conducted during the past few years and outcomes put forward in this thesis are mainly part of three larger research projects presented hereafter with their respective results. Although the scope of these projects might seem large, this provided insight into different aspects of quality in the electronics industry. More specifically, adopting different points of view on production-related problems allowed me to establish requirements for the quality improvement of processes rather than just products.

Concerning my own contribution, I would like to point out that all the methods and tools presented in the three case studies of this thesis have been developed or adapted directly by me, and their practical implementation has mostly been done by trainees under my strict supervision. For this reason, I am also the first author in all the publications related to these case studies.

TOOLMET. Within the "Applications of Adaptive and Intelligent Systems" program, my own research conducted in 1999 aimed at detecting placement problems and missing components on printed circuit boards (PCB). Research has been carried out in different ways to implement SPC-based hybrid methods in a data-poor environment. Emphasis was put on the responsiveness of the methods used for short-term data analysis. Furthermore, innovative methods to compress the already scarce data before statistical analysis were developed. Compression caused only minimal loss of information. This information could then be retrieved for short-term feedback control purposes. In this work, a low level approach has been used and line operators were identified as the main feedback recipients. Therefore, feature selection was specific to the level of information needed on the production floor. Case study No. 3 refers to this project.

INTELE. Taking place over 2001-2002, the aim was to detect defects during the final testing phase of PCBs and provide feedback to repair operators. After specifying the feedback needs, research focused on the analysis of functional testing data and definition of fault trees. Test design engineers were identified as the main knowledge pool for achieving traceability of the defect root causes. Emphasis was therefore placed on collecting this knowledge and creating an information gateway from the design stage to the repair stage. A rule-based system combining Linguistic Equations and expert knowledge as well as specific interfaces was developed in order to increase the interactions between users along the production process. Finally, reliable detection and localization of a defect was achieved with only the data equivalent to one day of production. Case study No. 2 refers to this project.

PRO-ELE. During 2003-2004, my research focus was on improving the controllability of the production tool. In order to achieve the target, a high level approach to the problem was adopted. Methods for knowledge acquisition as well as for automatic elicitation of that knowledge were developed. This can be used as enriched information for improving data analysis in a future knowledge-based decision support system. Emphasis was put on the involvement needed depending on the decision level in the company. Practical implementation was studied, leading to the development of an information-sharing concept providing added value to all users and aimed at improving their involvement towards successfully reaching the target. Case study No. 1 refers to this project.

1.6 General problem solving approach

A useful way for looking at processes is to use a functional point of view. A system can be seen as an association of sub-systems linked together to form a complex entity aimed at fulfilling a general function. In the same way, a sub-system is an association of components aimed at fulfilling one or several operational functions within the system. Finally, each component can be decomposed into one or several elements, each of which aims at fulfilling one single elementary function of the system.

According to the National Institute of Standards and Technology (NIST), a functional model is a structured representation of the functions, activities or processes within the modeled system or subject area. Methods such as IDEF (http://www.IDEF.com) or Structured Analysis and Design Technique™ (SADT™) were developed for this purpose. These methods are useful for system planning, requirements analysis, and system design. They were developed to provide a rigorous approach towards achieving understanding of user needs, prior to delivering a design solution.

SADT, for example, is a tool to structure information and allow engineers a better understanding of the problem to be solved through a common language and a scalable approach. By identifying elementary functions as well as their underlying mechanisms and environment, this approach enables development engineers to deal with each function separately. Problems can be solved at a local level because if a solution respects the local frontier, it is a sufficient guarantee for it not to create new problems at higher functional levels.

In the context of decision support systems, this approach is useful since DSS are often an agglomeration of different techniques and methods that aim at fulfilling one very general function. SADT diagrams have been used at different levels in this thesis, for designing the different tools presented in the case studies, but also more generally to define the overall structure of the thesis. The first case study defines the general framework whereas case studies 2 and 3 present more specific problems within this framework.

1.7 Outline of the dissertation

Section 2 introduces some issues and general considerations about quality management in electronics manufacturing as well as the concept of the knowledge-based decision support system. The following sections then present the three main axes of research. Each section includes a case study aimed at presenting and illustrating the author's own contribution to the topic.

Section 3 reviews the theoretical background as well as different techniques that can be used for knowledge extraction from various information sources in an attempt to improve the monitorability and the understanding of the production system. Emphasis is placed on the automatic elicitation of knowledge and interface adaptability that are needed to improve communication with the different knowledge pools.

Section 4 studies methods using operational knowledge for enhancing the comprehension of monitored production parameters as well as its effect on data preparation. It includes the cross analysis of data and knowledge based on fuzzy logic and linguistic equations and studies the ability of the methods to produce understandable diagnostics from the data analysis.

Section 5 deals with the level of online diagnosis that can be provided as well as its life span. Issues are addressed such as the formalization of results in order to provide comprehensive feedback information (user interface or automatic machine tuning) and self-tuning capabilities.

Section 6 discusses the research results and the methods and tools developed. It also verifies the hypothesis presented in section 1. The conclusions are given in section 7. Although the case studies relative to each section are presented in reverse chronological order, in the author's opinion this structure represents the most logical way to present the problem.

2 Related concepts and context

2.1 Decision support systems

The concept of a decision support system (DSS) can be broadly defined as a class of computerized information systems that support decision-making activities. Turban (1995) defines it 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". Although this might be a good definition, it is impossible to give a precise definition including all the facets of the DSS, as these are context- and task-specific. The only thing that can be said for sure about a DSS is that it includes a decision-making process, which has a more widely accepted definition.

Decision-making is the cognitive process of selecting one course of action from among multiple alternatives. It is said to be a psychological construct because the process of making a decision is not visible. Only the result of the process can be observed in the form of a commitment to take action. Decision-making is an important part of many professions, where specialists apply their expertise in a given area to making informed decisions.

Because decision-making is based on many different considerations, decision support systems belong to a multidisciplinary environment, including among others database research, artificial intelligence, human-computer interaction, simulation methods, and software engineering.

2.1.1 Different types of DSS

Depending on the level of support the DSS attempts to provide, there are three main categories:

- **Passive DSS:** assists the decision making process, but cannot produce explicit decision suggestions or solutions.
- **Active DSS:** can produce decision suggestions or solutions.
- **Cooperative DSS:** allows the decision maker to interact with the system by modifying, completing, or refining the decision suggestions.

At the conceptual level, the following further differentiation can be made (http://dssresources.com):

- **Communications-driven DSS:** It supports more than one person working on a shared task by using network and communication technologies to facilitate collaboration. In this type of DSS, communication technologies are central to supporting decision-making.
- **Data-driven DSS:** This type of DSS emphasizes access to and manipulation of a time-series of company data. Basic functionalities are provided by query and retrieval tools, but more complex systems allow the manipulation of data and even in-depth analytical processing of large collections of historical data.
- **Document-driven DSS:** It manages, retrieves and manipulates unstructured information in a variety of electronic formats by integrating a variety of storage and processing technologies. A web-browser including a search engine, for example, can be considered as a document-driven DSS.
- **Model-driven DSS:** This type of DSS emphasizes access to and manipulation of a model. A model-driven DSS uses data and parameters provided by decision makers to assist in analyzing a situation, but it is not usually data intensive.
- **Knowledge-driven DSS:** It can suggest or recommend actions based on specialized problem-solving expertise stored as facts, rules, procedures, or in similar structures. Additional knowledge is handled through data mining tools and intelligent methods.

2.1.2 Motivations for developing a knowledge-based DSS

People are an important factor in process improvement, because it is the people operating the process who are most likely to have the best ideas for its improvement. An operator who watches a process all day long will know how to improve it, as long as he is asked in the right way. For a long time the trend in process improvement has been to motivate operators to do their work in a better way. This is a waste of time, as motivation does not improve output permanently.

One solution for long-term process improvement is to integrate human knowledge into advisory systems. Seabra Lopes & Camarinha-Matos (1995) propose such a solution by describing the planning strategy and domain knowledge for error recovery. In their case, knowledge is obtained through machine learning techniques, some of which are described in section 2.

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) propose a framework for modeling such qualitative process characteristics by developing an intelligent computer-aided manufacturing system that can capture both the qualitative and quantitative aspects of a manufacturing process.

According to Özbayrak & Bell (2002), the development of knowledge-based DSS is justified by the inability of decision makers to efficiently diagnose many malfunctions, which arise at machine, cell, and entire system levels during manufacturing operations. Their knowledge-based DSS attempts to coordinate parts and tool scheduling activities and consists of three nested knowledge-based systems that aim to schedule the jobs according to the highest priority, provide a basis for selecting appropriate tools and hardware configuration on the basis of user-selected decision criteria, and provide an expert aid to diagnose tooling-originated manufacturing problems both at the design and at the operational level.

The examples given here show that for a DSS to be efficient, it should not rely only on one single type of input, nor should it rely on one single source. In the rest of this thesis, we will use the term “knowledge-based” for any system that relies at least partly on human knowledge without necessarily being driven by this knowledge.

2.2 Knowledge-based systems

According to the encyclopedia, data is thought of as objective facts. Data is the raw material for creating information that by itself carries no judgment or interpretation, no meaning. Information, on the other hand, is generally considered to be an advanced form of data in the sense that it is data that is organized, patterned and/or categorized. Dixon (2000) describes information as data that has been sorted, analyzed and displayed, and is communicated through various means. Generally speaking, information changes the way a person perceives something, thereby affecting judgment or behavior.

Knowledge refers to what one knows and understands. In the industry, it refers to the sum of information relevant to a certain job. In other words, it is what people have to know to be able to perform a job, for example the ability to recall the underlying principles and theories for a design engineer, or the names of the tools, resources, and procedures required for performing a task by an operator. Knowledge is richer and more meaningful than information. It is the internalization of information, data, and experience and can therefore be seen as the result of a learning process. Dixon (2000) describes knowledge as "meaningful links people make in their minds between information and its application in action in a specific setting."

The differences between data, information, and knowledge can be illustrated through the following example from the Free On-line Dictionary of Computing:

- 1234567.89 is data.
- "Your bank balance has jumped 8087% to $1234567.89" is information in the sense that a context has been added to the initial number.
- "Nobody owes me that much money" is the result of reasoning upon the given information. It is therefore knowledge.
- "I'd better talk to the bank before I spend it because of what has happened to other people" is an extra. The average person would consider it wisdom.

As a summary, the differences and most of all the gradation between data, information, and knowledge could be formulated in the simplest terms by the following:

Data	=	**Unorganized Facts**
Information	=	**Data + Context**
Knowledge	=	**Information + Judgment**

One might argue about this definition that it does not provide a method to make a clear difference between the concepts introduced in this section. The differences are mostly a matter of degree or point of view, and one person's knowledge can become another's information.

2.2.1 From explicit to tacit forms of knowledge

Michael Polanyi (1966) mentioned, "We can know more that we can tell". Knowledge that can be expressed in words and numbers only represents the tip of the iceberg of the entire body of possible knowledge. Knowledge is sometimes categorized as either unstructured or structured, and explicit or tacit. The transition between the different categories is fuzzy, which means that there is a continuum between explicit knowledge on one end and tacit knowledge on the other end.

If knowledge is "what we know", then explicit knowledge could be defined as "what we know we know". Explicit knowledge is formal and codified. It can be laid out in the form of scientific formulae, procedures, rules, principles, steps, and checklists. It has therefore two major advantages. The first is that explicit knowledge is relatively easy to capture and store in databases, computer files, documents or other hard copies. As a consequence, the second property of explicit knowledge is the ability to share it with a high degree of accuracy. Explicit knowledge can be either structured or unstructured.

- If individual elements are organized in a particular way for future retrieval, for example documents or spreadsheets, making it easy to share, then it is called structured knowledge.
- If the information contained is not referenced for retrieval (*e.g.* e-mails without subjects), then it is called unstructured knowledge.

Knowledge that is unstructured and understood, but not clearly expressed is tacit knowledge. According to Pan & Scarbrough (1999) "Tacit knowledge is not available as a text… It involves intangible factors embedded in personal beliefs, experiences, and values." This implicit form of knowledge could be defined as "what we do not know we know". Tacit knowledge is informal and uncodified. It is the knowledge that people have acquired through experience, sometimes internalized over a long period of time, during which it has been processed by their subconscious. This knowledge is carried in the mind and is difficult to access. Often, people are not aware of the knowledge they possess or how it can be valuable to others. Because it incorporates a lot of additional learning related to individual behavior and perception, it is also difficult to separate tacit knowledge from its possessor.

Tacit knowledge is considered more valuable because it provides a context for people, places, ideas, and experiences. Effective transfer of tacit knowledge generally requires extensive personal contact and trust. To convert implicit knowledge into explicit knowledge, it must be collected and formatted. Different methods for collecting knowledge are presented in section 3.1.2 and some of them are used in Case study No. 1.

2.2.2 Experts and expert systems

An expert is a person with special or superior skill or experience in a particular area who is widely recognized as a reliable source of knowledge in that area. Because of their prolonged or in-depth experience through practice and education in a particular field, experts' judgments are accorded a high level of authority and credibility by the public or their peers. The main difference between an expert and a technician lies in the amount of knowledge the first one possesses compared to the second one. Expert knowledge goes significantly beyond any general or shallow appreciation. This comparison, however, depends very much on a given context. Experts are usually needed to solve a certain problem or answer a specific question. The requested level of expertise will then depend on the level of the question. The status of expert is a context-related status rather than a universally agreed status.

In some fields, the definition of expert is well established by consensus and therefore it is not necessary for individuals to have a professional or academic qualification for them to be accepted as experts. 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.

"An expert system is regarded as the embodiment within a computer of a knowledge-based component from an expert skill in such a form that the system can offer intelligent advice or take an intelligent decision about a processing function. A desirable additional characteristic, which many would consider fundamental, is the capability of the system, on demand, to justify its own line of reasoning in a manner directly intelligible to the enquirer." This is the definition given by The British Computer Society's Specialist Group on Expert Systems (BCS SGES). Two characteristics can be retained from this definition.

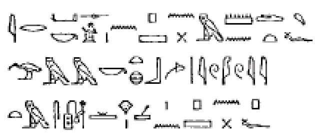
The first characteristic of an expert system is that it contains extensive knowledge on a specific field. To design an expert system, one needs a knowledge engineer, an individual who studies how human experts make decisions and then translates the rules into terms that a computer can understand. In other words, knowledge acquired from experts is categorized in a knowledge base. This is easy to do if the necessary knowledge is explicit, but sometimes needs a lot of translation work when the knowledge is tacit. Knowledge elicitation methods will be presented in section 3.1.3.

The second property of an expert system is that once the knowledge base is sufficiently rich, it can perform a task that would otherwise be performed by a human expert. In practice, expert systems perform both below and above the level of a human. However, in cases like medical diagnosis that involve mainly explicit knowledge, these systems often outperform human experts because of their systematic and methodical approach. It derives its answers by running the knowledge base through an inference engine.

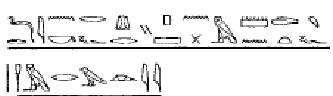
The idea of expressing and structuring knowledge in such a way is not new, since the oldest document that can be found on this topic dates from XVII BC. This is an old Egyptian text describing 48 surgical observations for head wounds, all of which are presented with a similar structure shown in Fig. 1: Title, Examination, Diagnostic, and Treatment. Other examples of knowledge bases can be found in ancient literature. These were usually highly structured, often of considerable size, and done by people motivated by the desire of keeping a record of their knowledge for future generations.

Nowadays, these motivations have not changed much, but the underlying technology has dramatically improved and extended their use. Some expert systems are now designed to take the place of human experts, while others are designed to aid them. These systems have a proven track record for solving diagnosis problems. But even if such a system is a form of artificial intelligence that uses a knowledge base of human expertise for problem solving, its success is based on the quality of the data and rules obtained from the human expert.

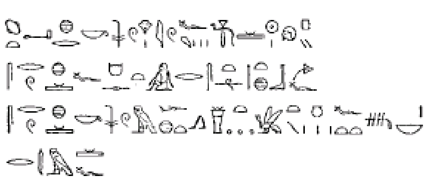
Title:
Instruction to heal a crack on a head bone.



Examination:
IF a man has a crack on one of his head bones, THEN you will find a red inflammation around this wound.



Diagnostic and prognostic:
You will say about him: “This is a crack of a head bone. It is something that I can heal”



Treatment:
You will apply fresh meat on the wound the first day. Then you will treat him with strawberries, honey, and bandages that have to be changed every day until complete healing.

Fig. 1. An early expert system from Egyptian times (translated from “Cahier Technique Merlin Gerin n° 157”).

2.3 Production equipment

Usually, the typology of production systems in the electronics industry consists of three main groups of manufacturing shops, which may be separate divisions of the global company or, as is the trend nowadays, totally distinct companies.

The first group consists of electronic component manufacturers that produce either integrated circuits or printed boards to supply PCB assembly factories. They are characterized by the use of cheap material but expensive production tools, and need therefore to maximize capacity.

The second group is represented by electronic subsystem assembly suppliers. These are usually subcontractors of larger companies, which produce, on a contract basis, PCB or other basic assemblies. Their processes are characterized by a high added value between the incoming material and outgoing complex subsystems.

Finally, the electronic end-product assembly suppliers represent the third group. They are the main order givers and the last step before the release to market. Their assembly operations are characterized by cheap and flexible production tools, but relatively expensive materials.

The focus of this thesis is mainly on the second group, because this is where the product functionalities are created. As a result, electronic subsystems assemblers are traditionally considered as a vector for integration of new technologies and are used by end-product assemblers as technological and quality buffers.

2.3.1 The assembly line

A production system is one that transforms raw material into something more elaborate by adding value to the product. In electronics manufacturing, these production systems are mainly composed of assembly lines, hence the added value could be defined as "the creation of required functionalities through the addition of different components and material parts". At least partly, the amount of added value depends on the technologies used for assembling the product.

A production system should not be reduced only to its physical part or a certain technology. Just as the fastest boat goes nowhere without a captain, no production system would be complete without a decision sub-system that controls and manages the flow of information and supervises the physical part. Whether material or simply information, flow control has become a necessity for companies that aim at improving their competitiveness and productivity. Nevertheless, achieving these targets involves a high level of flexibility from the equipment point of view. This task is challenging, since many parameters (factory layout, number of machines, automating level, flexibility of each machine, amount of operators, buffers etc.) can have an influence on the flows and it is therefore difficult to control them.

2.3.2 Manufacturing resources

Assembly lines have a linear design and are composed of a sequence of standard elements or machines linked together by conveyor belts as in Fig. 2. The main tasks fulfilled by the assembly line are:

- **Solder paste printing:** It is the selective disposition of solder paste over a surface using a screen or stencil, which has openings matching the pattern of the PCB. Applying the correct amount of paste at the desired location on the substrate is essential, as too little paste can cause open-circuits, while excessive paste can cause bridging of adjacent solder joints after the reflow process.
- **Component placement:** This is accomplished with pick-and-place machines. These machines can operate at high speed. The placement precision decreases as the speed increases, especially with larger and heavier components. Another factor affecting placement precision is the component shape, as the machine needs to establish component orientation each time it is picking it up.
- **Soldering:** Wave or reflow are the two soldering techniques used in PCB manufacturing. This is the component fixing stage, after which rework and repair operations will request a specially trained workforce.
- **Quality control:** Several testing methods exist and will be presented later in this chapter. The choice of the testing method depends on the expected fault spectrum.
- **Rework:** Most of the time, this is an end-of-line task, except if testing takes place at an early stage of the manufacturing process.

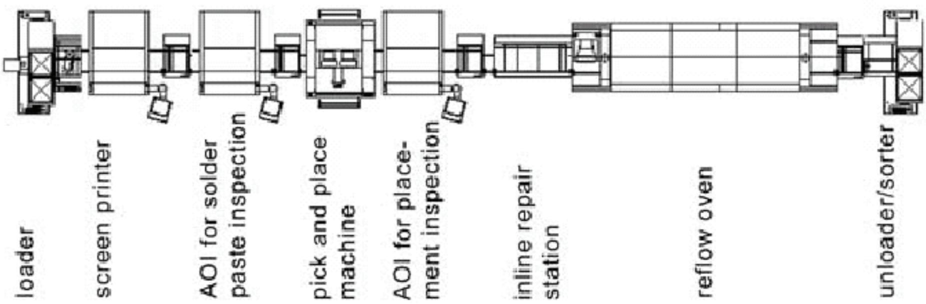


Fig. 2. Generic electronics assembly line (Kokott J, 2003).

Even if modern production systems are becoming increasingly automated, human operators still play an important part in production. Maintenance and setup of the production tools are tasks that rely on the human workforce. Depending on the type of production, a prompt reaction of the operators to fulfill their tasks is essential in order to have a smooth flow of material and avoid stoppages. A prototype production type is of course heavily dependent on human resources because of the non-negligible time spent in setting up the system at each change of production. But even in case of mass production, constant maintenance is needed due to the increasing complexity of the products and a high number and variety of components used in modern electronics. For all these reasons, human workforce still has to be considered as an independent resource and its scarcity or affectation policy will have a major impact on production.

Furthermore, even if quality control has been automated, human presence is often requested by inspection systems, whether it is to supervise or to intervene in problematic cases. Automated inspection requires an operator to select the programs, detect defects, and perform re-work on failed PCBs.

2.4 Monitoring and testing

Compared with other types of production, the special problem of electronics production is the large number of consecutive production steps with strong inter-relationships. The fact that many components are assembled onto one board leads to the impossibility of having zero defects. Because the correct function of the assembly normally depends on the correctness of all placed components, small connection errors usually lead to the failure of the whole assembly.

However, as end-users want to rely on fault-free products, it is important to have ways to test, detect, and localize faults efficiently. Two types of tests can be used for this purpose: structural tests and functional tests.

2.4.1 Structural testing

In-Circuit Testing (ICT) has been the most widely used inspection method for testing PCB integrity since the early 1980s. It is an electrical tester that stimulates a component or group of components with an electrical current and measures its value. This is achieved through the use of a bed-of-nails.

ICT attempts to isolate components and test them individually. Hence, it is highly accurate with a very low percentage of false alarms. Depending on test access, it can provide information about the specific component causing a problem, thus the fault report and debug time are small. Furthermore, ICT tests for component values and can show the presence or absence of components, misalignments resulting in open circuits, open circuits, short circuits and the correct operation of components.

The test generation is relatively easy because the circuit is tested one component at a time. Among several problems pointed out by Suck (2001), this raises the accessibility issue inherent to most of the structural testing approaches. ICT requires advanced equipment and physical access to all nodes. Modern assembly techniques make it almost impossible to access all circuit nodes on a PCB, and the test coverage of ICT is generally less than 100%.

The Manufacturing Defect Analyzer (MDA) is a cut down version of the ICT. Both MDA and ICT test for component presence, shorts and opens. An MDA does not power up the circuit, thus it has only limited capability to check component values. Fault diagnosis less advanced and the MDA can usually only point to an area for a fault instead of component level localization. One major advantage is that the cost of an MDA is usually about 25% to 50% of the cost of ICT.

Currently, most test methods relying on external test equipment physically accessing chips or boards are becoming increasingly problematic, as the integration density is getting higher. The Built-In-Self-Test (BIST) is an attempt to solve this accessibility problem.

The basic idea of BIST is to design a circuit so that it can test itself with minimal interaction from external test equipment, and report upon the test result. This generates a series of new challenges, starting with the necessity to make sure the chip is not lying. Another major concern, as explained by Pateras & McHugh (1997), is the additional circuitry needed to incorporate these new self-testing functionalities.

BIST presents many advantages over other structural testing methods. It provides hierarchical testing approach and can perform diagnostic at a variable resolution. Furthermore, even if test design costs are high, unlike external test equipment, a one-time investment in BIST stays with the product throughout its complete life cycle. Hence, BIST is by nature re-usable. Another advantage lies in the fact that it provides "at-speed" testing of the circuit and enables the reduction of software complexity due to the removal of functional test routines in the operational software.

Manual Visual Inspection (MVI) is the first of a sequence of defect filtering and screening steps at the heart of the classic inspection philosophy. It can be performed as a random control along the line or at defined work stages; for example, to double-check the results of an automated system that is not reliable as in Case study No. 3. However, MVI is subject to the limitations of human inspection, and lacks in repeatability. Typically, MVI covers only 50% to 80% of all defects, is inconsistent, and may cause unnecessary rework. Experience shows that manual visual inspection is the least effective and also the most costly approach.

Automated Optical Inspection (AOI) systems were first introduced to the PCB assembly industry in the early 1980s to replace human inspectors in the task of inspecting PCBs for defects such as missing parts, placement and similar errors. These first systems were expensive, slow and difficult to program. Today, AOI systems have overcome many of these limitations and can now be used at many stages of the process.

AOI essentially automates MVI. Each component in turn is viewed through a camera and is checked mainly for presence and orientation. AOI can also be used to inspect solder joints. A common criticism of AOI is that there are numerous false alarms. Whatever the aim of testing, PCB inspection algorithms fall into two approaches:

- **Reference comparison:** A scanned image of each board is compared to that of an ideal board to identify defects. This method, used in Case study No. 3, often produces a high rate of false alarms as the ideal board is usually replaced by a standard board that is supposed to be within control limits. Hence, the reference board usually already contains flaws.
- **Design-rule verification:** Inspection is simplified to a dimensional verification, which can be obtained from reliable sources such as CAD or CAM files. The advantage of this approach is that the time for program generation may be reduced from several days of manual teaching of patterns, to some minutes of program generation.

Compared to human inspection, AOI offers several benefits. Besides reducing the number of trained human inspectors, it also reduces the number of locations that a human inspector must examine to a few areas on the board. AOI also improves testing performance as it has much higher accuracy and repeatability than a human inspector, especially on dense boards. In these cases, only an automated inspection system is able to provide accurate and immediate feedback required to control and improve the process.

Automated X-ray Inspection (AXI) is an extension of AOI as X-ray inspection can also detect the presence, absence, or misalignment of components. However, its strength is in its ability to see through boards and components. When an X-ray passes through a board, dense areas such as solder will absorb more of the beam than the raw PCB substrate.

AXI is therefore mainly used to test the structural integrity of solder joints, including hidden joints that cannot be seen by AOI systems. However, X-ray inspection is a slow test method and thus not very suitable for in-line testing. It is normally used to perform sample-based off-line verification. Interpretation of AXI results also necessitates a high level of skill and experience, or advanced inspection algorithms.

For these reasons, AXI cannot be used alone for board inspection, but should rather be seen as a complementary technology to AOI and ICT. Together, these three methods have a much wider fault coverage than any of them individually as shown in Fig. 3.

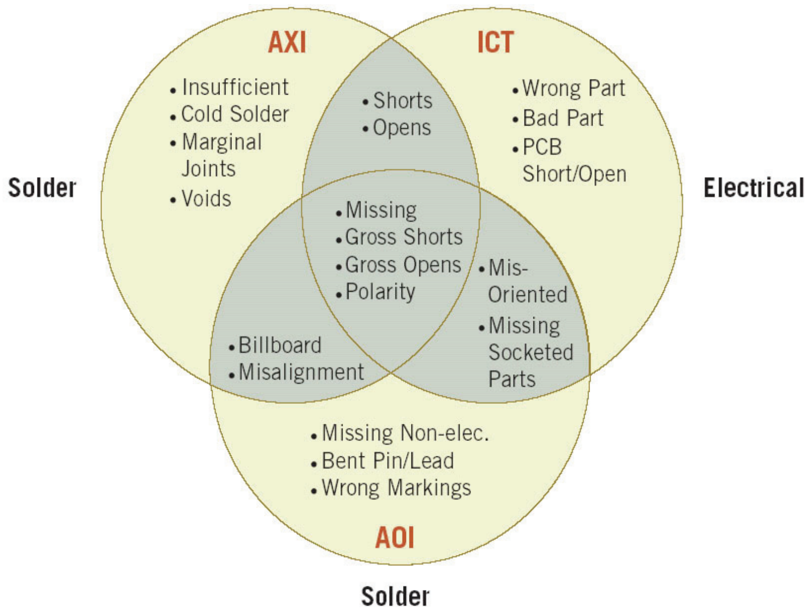


Fig. 3. Different testing methods can have an overlapping fault spectrum (www.OnBoard-Technology.com).

2.4.2 Functional testing

We have seen that the increasing complexity and density of modern electronic products makes testing via structural approaches very difficult. Only X-ray methods have sufficient resolution to provide accurate measurements. However, AXI does not provide enough fault coverage to be used independently. Consequently, functional testing is nowadays often added to structural methods, as component level access is not necessary.

Functional testing of a circuit is done by exercising its designed functions as accurately as possible and checking the subsequent dynamic behavior of the outputs. What is verified is the circuit's proper functioning in an environment similar to the one it has been designed for, but this does not require any other connection as the board's connectors. Faults will be detected by analyzing the board's functional response and timing aspects.

Functional testing is a complex approach and a lot of research is still required if it is to become an efficient inspection method. Currently, the major drawbacks are:

- **Test program complexity:** Although it can be applied in almost any environment, the test software complexity is generally very high and case-specific. This requires long test program development and expertise in the field of functional testing.
- **Fault localization:** As components are not tested individually, fault localization is much more complex and requires a trace-back methodology.
- **Fault coverage:** Functional testing does not provide 100% coverage, which is limited by the fact that there is no component level diagnostic.
- **Employee collaboration:** Unlike structural testing methods that can be developed by local specialists, functional testing requires highly skilled technicians who, in their turn, rely very much on feedback from the field for their diagnoses.

For all these reasons, companies might be reluctant to use functional testing, or at least mistrust the test results. It would be a shame, however, not to use the methodology because of its apparent complexity. On this topic, Case study No. 2 presents an example where functional testing was used for improving rework operations efficiently.

Functional testing can also be applied to final products, whether it is the last step before release to market, or a diagnosis tool for products that were returned by customers. This approach is already used by car manufacturers when running check-ups on recently supplied vehicles. Maintenance is limited as the inspection system will diagnose the problem and provide repair indications.

2.5 Quality and the electronics industry

Electronics production is determined by extensive use of testing and repair operations that represent a non-negligible part of operative costs. According to Feldmann & Sturm (1993), these costs represented in 1993 already up to 50% of the total costs of a product and have since been increasing with the complexity of new products (Feldmann, 2003). There seems to be the belief that in order to achieve quality, everything must be checked, tested, inspected or measured. This attitude toward quality dates from the 1930s.

Nevertheless, important process variables are often left out. Many companies do not use tests as they should and miss the knowledge gain and potential improvement that could be made to the process.

The first reason why important process variables might not be recorded is because they are not always measurable. However, Section 3 of this thesis will show that methods do exist to solve the problem of approximate and incomplete databases. There is a second reason which is more worrying. Electronics and especially the telecommunication industry tend to focus more often on verifying a product's functional quality rather than on insuring the operative quality of the process. In a sense, it is as if a century of evolution in quality control methods went totally unnoticed by some of the players in the electronics industry, especially among subcontractors.

This might seem amazing in a sector considered by most to be at the forefront of innovation and a symbol of high technologies. It is true that research and development has traditionally been focused on improving the performance of the product and relatively little attention has been given to the efficiency of the production. Spanos (1992) suggests that the rapid growth in information technologies is to be held at least partly responsible for this attitude. Heavier industries such as automotive or aeronautics have learned to survive in a competitive environment and mentalities have evolved and adapted to markets where it is best to produce what has already been sold. The electronics industry has seen exponential growth of its markets in a little more than a decade, hence the context was very similar to the Taylor system where almost every produced item found a buyer and margins were high. These high value-added products did not encourage either the industry to undertake rationalization or quality improvement policies.

Today, the situation has changed and the environment has become more competitive. In electronics, this change has been brutal, whereas in other industries it has been taking place sometimes over several decades, allowing the time for a new generation of quality engineers to arise. Moreover, quality improvement measures have a wide impact on the production culture and are therefore as much a managerial challenge as a technical one. Many companies are still struggling with cost-cutting measures. One could say that the dinosaurs of the former generation are not extinct and their mentality has not yet fully integrated or understood the new challenges of this environment.

This is again most flagrant among subcontractors where many quality assurance implementations are a result of an external requirement. "Our biggest customer asked if we had ISO9000" is a typical reason for implementation. This is more rarely followed by the proper involvement at every level of the company throughout the year, instead of just during quality audits. Consequently, shop floor quality inspection still primarily concentrates on separating "good" from "bad" quality. Progress in yield improvement is rather difficult to achieve if the link between information coming from the inspection of product features and process data that lead to good or bad quality is missing. Quality control mechanisms are characterized by open loops based on low-level heuristic knowledge of the operator. This means that when quality problems occur, normally the reasons that caused the problems cannot be assigned because the relevant process data are not known.

2.6 Conclusion

Testing and repair operations represent a non-negligible part of the costs in electronics manufacturing. Although different approaches exist, each methodology has its own weaknesses. No single approach allows complete fault coverage and identification may even become impossible until after a field return. Customers represent the ultimate control stage, but waiting for them to detect a defect can have much worse consequences.

In order to avoid such situations, multiple inspection approaches should be used, both at the structural level and the functional level, as their fault spectrum may overlap. Furthermore, an efficient fault inspection also requires a company level involvement and a change in mentalities. In some organizations, test equipment is often used as a limited fault detection system. Hence, it is a way to find faults to give to technicians to fix.

As the issues and limitations of individual testing approaches show, the purpose of the test is much larger. When used correctly, it can provide feedback to the processes to prevent the fault from ever occurring again. This requires inspection systems that are not only able to read data and check limits, but also to produce understandable information that will be distributed to the relevant people.

3 Defect-related knowledge acquisition

3.1 Acquiring the knowledge

One of the major issues encountered when building knowledge-based decision support systems is acquiring the necessary knowledge or expertise. Unlike data mining, which is a well-studied topic, knowledge acquisition is often not given enough importance. It is quite common in electronics manufacturing to obtain large databases and run blindly feature extraction methods. But data does not exist naturally in a factory; first it has to be collected, stored, and prepared as explained in Pyle (1999) and Weiss *et al.* (1998).

Combining this data with knowledge of the process could drastically increase the performance of these tasks. Knowledge acquisition is an essential stage of the knowledge engineering process. During knowledge acquisition, the knowledge engineer carries out a two-step process. Extracting the knowledge from the expert is the first. Representing it in a knowledge base is the second.

3.1.1 Knowledge bottlenecks

Identification of useful features needs knowledge about the process. Luckily, unlike data, knowledge does exist naturally in the factory, but collecting and interpreting this knowledge is a very difficult task. It is often referred to as the bottleneck in the expert system development process as Feigenbaum & McCorduck (1983) explain. Some of the major issues that can be encountered during the knowledge acquisition process are listed hereafter. These issues are divided into three main categories.

Most knowledge is not only in the heads of experts, but it is also tacit. It is referred to as the “knowledge engineering paradox” by Liebowitz (1993). This paradox states that the more competent domain experts become, the less able they are to describe the knowledge they use to solve problems. Experts tend to explain their reasoning in general terms that are too broad for machine processing. They combine pieces of basic knowledge together in an unaware and non-sequential process that is difficult to describe.

And when they have to examine a problem and explain their conclusions, it is also therefore probable for them to forget the smaller steps and only give the main ones.

In addition to having mostly tacit knowledge, experts also have vast amounts of knowledge. Studies have shown that a single day of knowledge acquisition can produce 300-500 pages of knowledge elicitation transcripts, and one day of the expert's time is usually needed for every four days of the knowledge engineer's time. Not all this knowledge, however, is relevant to a given problem. The way knowledge is being collected can also induce a bias (*e.g.* when questions are too specific).

It is often said that knowledge is power and some people are reluctant to share it because they fear for many reasons that they are relinquishing their power. An expert can fear disappearing in the mass of expandable workers if he gives up what makes him unique in an organization. Verkasalo (1995) states that "the success of knowledge management initiatives depends upon people's willingness to share knowledge and use the knowledge of others" suggests motivating knowledge sharing by giving people credit for their ideas. It is a way to alleviate problematic interpersonal communication factors that may exist between the knowledge engineer and the expert.

Another reason for not wanting to share knowledge can be related to company policy as companies have ever-stricter rules about confidentiality. My own personal experience showed, however, that confidentiality may sometimes also hide a lack of expertise. In particular, subcontracting companies are not always given all the useful information by their clients or do not have sufficient resources to afford a particular kind of expertise.

Depending on the level of expertise required to solve a problem, finding an expert can be a very difficult task. Experts are not always known, their time is precious, and since most of the expertise is tacit, they might not even realize it themselves. Amazingly, however, as soon as his new status is revealed to an expert, he tends to become suddenly as busy and unavailable as his peers. A good knowledge management organization is therefore necessary to promote knowledge sharing and avoid scaring off experts.

Other problems of knowledge availability are linked to today's global working conditions. 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. But globalization also brings another change in working conditions as described by Verkasalo (1995). Hierarchical management structure moves towards process management and orders are not anymore given exclusively by upper chains of command, but rather by the front-line process teams. This autonomy can lead to individualist and isolationist behaviors. Distributed decision making becomes therefore a major issue. Knowledge must be easily accessible because decisions made at one place have an impact on the work of other teams worldwide. The "It's not anymore our problem…" attitude must be avoided because it goes against global process management.

3.1.2 Knowledge extraction

Knowledge bottlenecks call for specific techniques to acquire knowledge from experts. The main characteristics of these techniques are:

– Experts must be taken off the job for only short time periods.
– Non-experts have to be able to understand the knowledge.
– The focus should be on essential knowledge and irrelevant data must be filtered out.
– Tacit knowledge is the prime form of knowledge that has to be captured.
– Knowledge representation must allow knowledge to be collated from different experts.

Because there are many different types of knowledge possessed by experts, numerous techniques have been developed to help extract knowledge. Three ways have been studied to improve the efficiency of the knowledge extraction process:

– **Knowledge extraction methodologies:** They are intended to provide guidelines for systematic knowledge extraction activities, whether these are performed manually or automatically.
– **Ontology:** The term ontology refers to an explicit and formal specification of how to represent objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them. Finding the right ontology can also be referred to as the knowledge elicitation activity.
– **Software tools:** Their increasing use aims at assisting the extraction process. In case of automatic knowledge acquisition, knowledge extraction methodologies are often embedded within the software tool.

Although this thesis focuses on collecting knowledge automatically, manual extraction techniques also have their importance and should be kept in mind as they form the base for many automatic acquisition methods. For example, expert interviews are still the most widely-used method. The main advantage is the ability to make tacit knowledge more explicit by allowing the expert to provide a context and explanation to his actions. When using multiple sources of expertise, Turban (1991) suggests defining primary and secondary experts for solving problems linked to conflicting opinions.

The drawback of manual extraction is that the efficiency of these knowledge extraction processes depends very much on the knowledge engineer's own understanding of the domain. This is why experts should be able to be their own knowledge engineers and collect their own expertise into knowledge bases.

3.1.3 Knowledge elicitation

Communication problems between the knowledge engineer and the expert make it difficult to transfer expertise into a program. The vocabulary used by the expert is often inadequate for problem-solving because it is not understandable by others and even less by machines. The ease of solving a problem is very much determined by the way it is conceptualized and represented. Ontology aims at sharing and communicating knowledge, both between people and between computer systems. It is a necessity when building knowledge-based systems because knowledge can only be stored and manipulated effectively if it is in an abstract form.

The model for knowledge engineering has long been that the knowledge engineer mediates between the expert and the knowledge base. However, knowledge conceptualization is a difficult task. While an expert will find it easy to demonstrate his expertise in a practical situation, he will struggle to formulate explicitly his knowledge as rules or other abstractions. Therefore, a representation of facts only becomes knowledge when used by a person or program to behave in a knowledgeable way.

Concept learning is the acquisition of structural descriptions from examples (MacDonald & Witten, 1989). It involves acquiring descriptions that generalize explicitly the structure of knowledge. For a person, learning is not simply a matter of acquiring a description, but involves taking something new and integrating it fully with existing thought processes. Computer programs using concept learning are not nearly so capable. However, they do acquire explicit description that can support different kinds of reasoning, for example in the form of rules. Two kinds of concept learning can be distinguished: inductive and deductive. The distinction between them is like distinguishing necessary from contingent truths. Necessary truths are facts that could not be otherwise, in other words they can be deduced from the domain theory. Contingent truths could be otherwise and hence cannot be deduced.

A number of generic knowledge representations have been constructed, each having applications across a number of domains. Holsapple *et al.* (1989), in particular, have worked on generalizing knowledge representation and elicitation by trying to answer the following three questions:

1. What are the different forms of knowledge?
2. What are the criteria in selecting knowledge representation forms?
3. How is knowledge to be elicited?

The first two questions can be addressed using generic tasks. For example, medical diagnosis and fault detection can both be categorized as classification tasks. The two problems, though different in their domain areas, are structurally similar in their use of diagnosis rules as seen in Fig. 4. Based on the notion of a generic task, it is then natural to have different generic knowledge acquisition methodologies. According to this mapping, knowledge representation depends on the task. Common classes of knowledge representations are:

- **Logic:** This representation is used by expert systems based on first order logic to prove a theorem. Relationships are stated as a collection of primitive assertions connected by logical operators (*e.g.* AND, OR, NOT). In these systems, a logical statement is a description of a concept. Interestingly, induction is the popular strategy used to learn a concept. Examples of a concept are stated in terms of logical statements. The generic approach is to generalize these statements to ones that are more descriptive and at the same time consistent with the given examples.
- **Semantic Network:** When using a semantic network, concepts are represented in the form of graphs that have nodes depicting instances, concepts (*e.g.* color) and attributes (*e.g.* blue), and arcs to represent relationships (*e.g.* is a).
- **Production rules:** Knowledge in the form of IF-THEN rules is a common scheme for representing reasoning knowledge. Production rules have a similar form to statements in logic. However, there are basic differences between the two in their inference mechanisms.

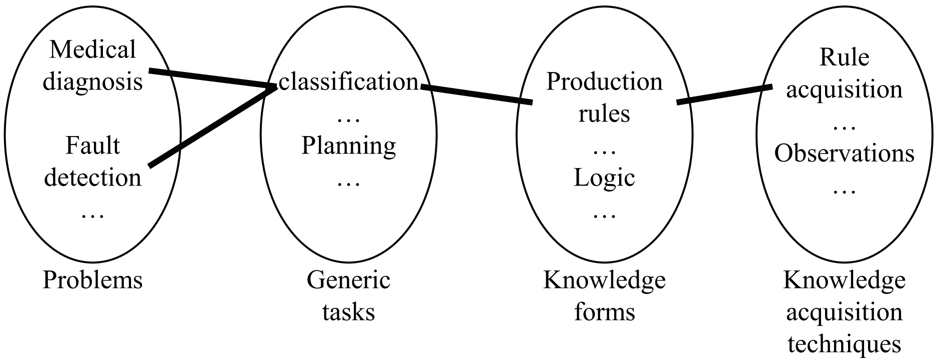


Fig. 4. Problem-Task-Representation-Acquisition mapping (Holsapple, 1989).

In order to structure knowledge acquisition activities throughout the design and development process of a KBS, Rook & Croghan (1989) propose a framework represented by a knowledge activity matrix. This matrix represents knowledge acquisition according to the following three major dimensions:

– **Time dimension:** The time dimension consists of the different phases in systems engineering. It goes from the initial definition and specification of a problem and its environment to the post-installation system maintenance.
– **Logic dimension:** This dimension includes the knowledge acquisition steps that are conducted throughout the system engineering phases (knowledge resource identification, macro-knowledge extraction, knowledge partitioning, micro-knowledge extraction, knowledge formalization, and knowledge encoding). It is in effect a refinement of the two-step approach (knowledge extraction and elicitation) put in context.
– **Environment dimension:** This represents the setting or environment that the completed system will be developed for and reside in (laboratory environment, simulated operational environment, and operational environment).

3.1.4 Automatic methods

The methods described in the previous sections have some limitations in common; they are slow, expensive and do not always give accurate results. Therefore automatic knowledge extraction methods have been developed during the last decade. In theory it is now possible to build a knowledge base with no need for a knowledge engineer and only very little need for an expert. Induction learning, for example, is a way to learn from experiments without having any predefined rules. A knowledge base is simply generated over time by the generalization of the accumulated specific cases.

Another example of advances made on computer-based knowledge acquisition is the knowledge acquisition dilemma: If the system is unaware of the context, it cannot raise good questions; if it is knowledgeable enough, it does not require raising further questions. On that topic, Kawaguchi *et al.* (1991) have developed an interview metasystem. It specifies question strategies by analyzing dynamically the nature of the task and raises further questions to elicit any new knowledge and refine the domain model. Gaines (1993) proposes elicitation tools that are computer-based interviewing techniques extended through the use of graphical rather than numerical data entry.

Based on the fact that many of the existing tools are too domain-specific, Chien & Ho (1992) propose an approach through interactive knowledge acquisition tools. The problems of restrictive knowledge categories and different knowledge representation are solved through a generic knowledge acquisition shell. Tecuci (1992) focuses on extending and updating incomplete and possibly partially incorrect knowledge bases through gradual development and integration of new input information received from the human expert. On a similar topic, Winter (1992) uses automatic knowledge acquisition to extract the missing knowledge and refine the database.

More generally, when developing expert systems, more than 50% of the man-hours for the whole task have been spent on knowledge acquisition. Okamura *et al.* (1991) have therefore developed a system for shallow knowledge, which is a heuristically acquired experience. The system automatically gathers a large volume of case data and generates rules in the form of a decision tree. Instead of suppressing the knowledge engineer as an intermediary, Fujihara *et al.* (1997) try to facilitate his work and compensate for his lack of knowledge. They present a domain independent knowledge conceptualization tool that supplements the knowledge engineer's lack of domain experience, so that the resulting knowledge base is accurate and complete.

In a domain closer to the focus of this thesis, Pierce *et al.* (1988) have developed an automated knowledge acquisition editor for diagnosis and repair problems. It permits expert technicians to work alone in specifying fault isolation and repair plans for specific electronics systems. The system features a graphical direct manipulation interface. Parts, tests attached to each part, and repairs associated with each test form a tree of components that the expert can traverse and specify dependencies. Diagnosis with expert systems can also be affected by irrelevant and in some cases even erroneous knowledge. For this problem, Su *et al.* (2002) propose an improved genetic algorithm that applies operators such as selection, crossover and mutation to evolve an initial population of diagnostic rules and optimizes it. This last topic will be seen in more details in section 4, in which a new solution to the problem of incomplete knowledge bases is proposed.

3.2 Building an interface

In Decision Support Systems (DSS) and more generally in many computer applications, 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. With hardware and applications becoming more complex, the area of human-computer interface technology is becoming essential. In this context, the user interface

(UI) can be seen as the dialogue component of a DSS that facilitates bidirectional communication between the system and its user.

3.2.1 Aim and description of a user interface

To work efficiently with a system, the user needs to be able to assess its state and control it through actions. Norcio & Stanley (1989) give the example of a car in which case the UI is the instruments the driver can act on or get information from, in his attempt to accomplish the tasks of driving. This example shows that the term *user interface* is not limited to computer systems and electronic devices, but is more generally part of a man-machine system (MMS) and a subgroup of the human-machine interface (HMI). An efficient interface is one that provides adequate interactions between the different participants of a MMS and allows information exchange. Bálint (1995) defines the characteristics of such interactions. Adequately, they should:

- facilitate satisfactory monitoring of machines to humans;
- support human interventions into machine operations;
- aid human decision making by providing for example system state diagnosis and intervention possibilities;
- establish error-free or error-tolerating operation of the full system, and by this way
- guarantee efficient and reliable system functioning.

Currently different types of interfaces exist in the area of computer engineering. The most common types are:

- **Command-line interfaces:** Input is provided by typing a command string with the computer keyboard and the system provides output by printing text on the computer monitor. This kind of interface is widely used but shows poor usability and lacks explanatory power as will be seen a little further in the current section.
- **Graphical User Interfaces (GUI):** The GUI is an interface that uses a computer's graphics capabilities to make the program easier to use. It accepts various input devices (*e.g.* keyboard or mouse) making it possible to select objects, including icons, menus, text boxes, etc. A GUI also provides graphical output on the computer monitor, making it easier to represent conceptualized information.
- **Web-based interfaces:** Web-based interfaces are an extension of GUIs. They accept input and provide output by generating web pages which are transported via the Internet or a Local Area Network (LAN) and viewed by the user using a web browser program. They use the graphical and distribution properties of the web.

The choice of an interface will depend on many factors, such as the type of users, the environment or even the competence of the interface designer. Since interfaces are very context-specific tools, one could imagine different declinations for every single situation. For a given situation, however, not all choices are suitable. Norcio & Stanley (1989) define three major factors underlying the inadequacy of HCI technology:

- Interface software is often developed by software specialists without sufficient knowledge about the domain and is therefore not viewed as the part of the system but rather as a software package between the system and the user.
- The design of effective interfaces is a difficult problem with sparse theoretical background. Interfaces are objects lying between the domains of expertise and it is difficult to unify into one theory something which is resulting from a multidisciplinary approach.
- Software engineering principles are generally not given significant consideration in designing interfaces. Specifically, user specifications need to be incorporated in the design of human-computer interaction software for an improved usability.

3.2.2 Usability

The major focus of system design traditionally concerns the functionality of the system, but rarely how or by whom this system will be used. Norcio & Stanley (1989), on that problem, state that "the user interface, which is the component of the system that communicates with the user, is typically considered as the incidental part of the system and is frequently viewed as an afterthought". With this approach, however, both the interface and the user remain external to the system rather than parts of it. Moreover, the resulting interface is frequently not well suited to the system or to the user, and more often to neither. Designing software that is consistent as well as functional and easy to use is the central issue in software development and determines the product's success. Only team work will help developing real "user-friendliness" according to Wills (1994), who proposes an organizational structure for collaboration between the specialists of different fields involved in the interface design. This way usability issues will be constantly kept in mind during the whole development process.

Ergonomics, according to the encyclopedia, is the science that studies the interactions between products and their users and thus tries to maximize usability. The term "user-friendly" is sometimes used in software engineering when a product includes a UI. But this term refers more often to a marketing attribute than to real usability issues. "User-friendly" means more than just using a GUI and psychological ergonomics explores design issues in terms of cognitive psychology, cognitive workload, human error, human perception of their surroundings, and, very importantly, the tasks they choose to undertake. Usability is improved by answering questions such as:

- Who are the users, what do they know, and what can they learn?
- What do users want or need to do, what is the context?
- What has to be left to the machine? What to the user?

Answers to these questions can be obtained by conducting user and task analyses. Including them into the design of the interface will affect the amount of effort the user will have to spend to provide input for the system and to interpret its output, as well as how much effort it takes to learn how to do so. Usability can therefore also 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.

3.2.3 Adaptable versus specific interfaces

Knowledge-based systems are not always acceptable to the people they were intended for, and are certainly unsuitable for casual users. Incorporating user specifications in the design of human-computer interaction can improve usability but will in most cases remain limited to the generalization property of the common user. According to Holsapple (1988), typical user interfaces tend to be rigid and incapable of supporting diverse presentation modes for a given piece of knowledge or of adapting to changing circumstances. The user must conform to the interface, rather than the interface adapting to his needs. An ideal computer system, however, should adapt to the current user by compensating for weaknesses, providing help appropriate to the context, and decreasing the mental and physical workload of that particular user. Flexibility therefore has the potential to improve user productivity and effectiveness and, as a result, the speed and quality of the decision making process. Norcio & Stanley (1989) define two main ways in which a system can be adaptable:

- **Static adaptability:** The interface can be modified by the user if the behavior of the system is deemed unsatisfactory. Although this may produce a better interface, it leaves the burden of adapting to the user.
- **Dynamic adaptability:** In this case, the interface works differently depending on the current context. This includes both the current task and the current user. In the following part of this section, unless specified otherwise, the term "adaptability" will imply its dynamic form.

When investigating HMIs in the view of their flexibility to changing human properties and circumstances, a general rule appears. For an interface to be adaptive, it needs to include knowledge from at least the four following domains describing the current context (Norcio & Stanley, 1989):

- **Knowledge of the user:** An adaptive system must be able to characterize and distinguish between individuals and needs therefore a model for each of them. This model should reflect the user's knowledge of the system and the task domain.
- **Knowledge of the interactions:** For an interface to provide information appropriate to a given context, it must be able to characterize the current status of the human-computer dialogue. It must therefore be able to keep track of that dialogue, including the information that may be implicit in it.
- **Knowledge of the task/domain:** In most HMIs, the user is trying to achieve goals and the aim of the DSS is to assist him in achieving them. If the goals are implicit, the interface must be able to infer this information from the interactions.
- **Knowledge of the system:** In addition to knowledge about the user, an adaptive system should have knowledge about itself. The system must be able to optimize input and output according to its own current status and within the boundaries of its own limits.

Early user interfaces were developed based on user models that were static and used stereotypes. But in order to provide the user with understandable and credible information, the UI should make it possible for a DSS to tailor its response to the needs of the individual. Interface adaptability has therefore been the subject of many studies in the beginning of 90's. Jerrams-Smith (1990), for example emphasizes the need of acquiring dynamic user knowledge to build better interfaces and response. Eberts (1991) uses neural networks for mapping users' actions to what they intend to do. By learning what a user intends to do, this method can then serve as an information filter and provide the user with only the pertinent feedback. The same problem is studied in Lind *et al.* (1994), followed by Marshak *et al.* (1994), but this time using interactive concept mapping and cognitive engineering to match what a user wants to what he needs.

More generally, HMIs raise the problem of interfacing a machine designed by intention so that it behaves deterministically with a human who is behaving based on properties such as his physiological attributes, intellectual characteristics, knowledge basis, or psychological state. According to Bálint (1995), for achieving enhancements, adaptable interfaces must therefore be capable of adjusting the forms of information transfer, transforming the information content, altering modes of information flow, and exchanging or combining communication media

Adaptable interfaces, even if they do have the potential for improving HMI, are however no panacea. Norcio & Stanley (1989), for example, point out some negative aspects of an adaptive HMI. Besides an increase in implementation complexity and cost, adaptability might undermine the user's confidence in the information given to him. The user may not be able to develop a coherent model of the system if the system is frequently changing. For this reason, adaptability might also be applied to the content instead of the interface itself. A case study presented at the end of the current section illustrates this idea. Through the use of fully graphical interactions and digital pictures, it is possible to update or change the context within a given interface. It is therefore an attempt to adapt the embedded problem while keeping a fixed framework.

3.2.4 Explanatory power of an interface

Traditional interfaces are not only too rigid, but very often they do not explain very much and are intolerant to an end-user's need to understand the way a KBS is generating its recommendations. This need to understand raises two questions:

- How can a KBS explain itself, meaning what form should this explanation take?
- How much explanation is it wise to provide to the user?

These questions introduce the concept of the explanatory power, which refers to the ability of a KBS to explain itself. Two related characteristics are relevant to understanding explanatory power: transparency (the ability to see the underlying mechanism of the reasoning process) and flexibility (the ability of the interface to adapt to a wide variety of end-user interactions). In order to improve explanatory power, Natkatsu & Benbasat (2003) propose two kinds of enhancements that can be made on a KBS, both of which have the potential to increase transparency and flexibility.

Content-based enhancements refer to the increasing of the information content associated with the reasoning process of a KBS. Traditional expert systems, for example, have often provided little or no explanations about their recommendations. Even when providing explanations, these were often inadequate, non-instructive, and difficult to understand. This can occur when user specifications are forgotten in the design of the KBS. Some interfaces, for example, contain the full underlying architecture of the KBS and are therefore more suited to the knowledge engineer than to the end-user. They are graphical representations of the undiluted rule-base and do not enable the user to grasp the concepts.

On this topic, Lambert & Ringland (1990) state: “one principle whose realization in a knowledge-based system would be highly desirable is that the system’s intelligence perceived by the user should closely match its actual intelligence”. This translates into the idea that a KBS should indeed provide the user with explanations about its conclusions and reasoning. These explanations should be neither too rudimentary nor give a false sense of certainty when that is not justified. The explanation given to the user should not serve as a selling argument for the decision, but should rather enable him to criticize the system’s reasoning. This is particularly important if the system is operating near the bounds of its expertise. Enhanced explanations can clarify the user’s view of the KBS’s internal reasoning, enabling him to assess the advice, understand the causes resulting in that advice, and eventually revise the recommendation if the user deems it necessary.

Interface-based enhancements focus on designing the user interface in a way that improves transparency and flexibility in the system. Adaptable interfaces can, to some extent, play a role in achieving this target. Explanation may be presented to the user in the form of graphics or text. However, a text lacks in flexibility and in non-standard situations it might suggest to the user more than it actually means. Furthermore, knowledge bases contain mostly explicit and structured knowledge and are well suited for graphical representations. Some KBSs are therefore using some form of graphical interface, allowing the user to explore the knowledge base.

Natkatsu & Benbasat (2003) propose to use graphical hierarchies instead of the equivalent flat interface to describe the structure of a rule base and its relationships. Fig. 5 illustrates this idea for a delivery company having to choose the transportation means according to various inputs. This representation is especially well suited for expert systems or more generally any rule-based system since they use hierarchical models. These systems rarely display this information and rather leave the reasoning process invisible. The graphical hierarchical representation, instead, is intended to be interactive and allows the user to inspect parts of the model. He can then better understand why the KBS had reached a certain conclusion and, if he disagrees with it, he may want to track the input variable that was the cause of the outcome.

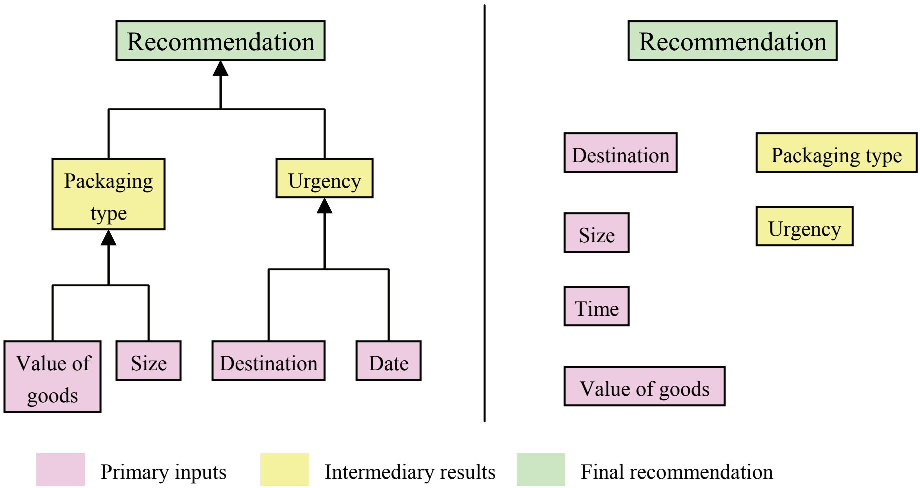


Fig. 5. Simplified hierarchical model and its equivalent flat model for a delivery company.

3.2.5 Design Process

Traditional software development follows the "waterfall" software life-cycle paradigm, which includes six basic stages: requirements, specification, design, coding, testing, and maintenance. This method is not adequate when developing a user interface since it focuses solely on the functions of the software being developed and the data on which they operate, whereas interfaces serve as a bridge between these functions and users and are intertwined with functional parts. Xing & Li (1992) attempt to analyze the inherent dialogue between the user and the computer, and build up a widely applicable model to guide the design of the human factor aspect of software. This can then allow the design and implementation of user interfaces independently of functional requirements. However, the efficiency of an interface is difficult to maximize when using a generic method and interfaces need specific development methodologies.

The user interface design process can be defined as the total set of design tasks to be carried out to transform user requirements into user interface design specifications. UI design requires a judicious mixture of creativity, design knowledge, experience, task analysis and a thorough understanding of users' requirements. It is important for engineers and other members of the development team, not only UI specialists, to be familiar with the important concepts of UI design because the development process is traditionally driven by the designer and developer. Wills (1994) proposes a work organization that tries to promote the involvement of the whole panel of specialists during the entire development process of the interface so that engineers understand that the interface is more than the devices with which the user interacts with the system.

More generally, the best way to ensure that a system is friendly and works is to base it on the intended user's mental model. Because designers do not spend enough time thinking about the design problem, Balasubramian *et al.* (1998) introduce the need for a formal planning and idea generation phase (creative phase) in the UI design prior to prototyping. They also propose a systematic design methodology containing fifteen non-sequential, highly connected, and iterative set of design tasks to structure the creative thought processes of the designer with appropriate guidance. This methodology is aimed at reminding UI specialists that their interface is to be the part of a larger system. The user-centered approach to interface design for knowledge acquisition proposed by McGraw (1994) is more convenient for knowledge engineers. The following steps 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.

3.3 Case study No. 1: Expert Knowledge Acquisition System

3.3.1 Problem description

This case study has been the object of a publication (Gebus & al., 2004). The company involved in this case study is producing, among others, internal mobile phone antennas by assembling both mechanical and electronic parts. Although this does not relate strictly to electronics assembly as described in section 2.3, here the general framework for defect-related knowledge acquisition is defined. The case company, a subcontractor for some of the major players in mobile telecommunications and facing the same problems as many other subcontracting companies, provides a suitable application area.

Client companies and order-givers are mainly interested in getting products in-line with the specifications that they define in their own design centers, unfortunately too often rather regardless of manufacturability concerns. As a result, the quality problem is shifted to the subcontractor who must find the right balance between long-term and short-term concerns; investing resources in the development of operative production quality on one hand, and assuring it already now for their customers on the other. Quality assurance is usually very well taken care of by the quality department since the first priority of a subcontractor is to satisfy its customer, meaning to produce within given specifications. The problem arises when these specifications are solely linked to the functionalities of the product, because these can be totally disconnected to any manufacturing method. In such a case, emphasis is on product quality at the expense of process quality that becomes ever more difficult to control. However, the two concerns mentioned above are not contradictory and juggling with both of them should not be impossible.

The proposed system tries to solve this problem through the development of tools that would allow the company to understand and formalize the different parameters influencing the product quality, hence to improve the operative quality through feedback control.

Operators 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. Since operators are not yet available in standard format! A decision support tool could lower the variability and greatly improve the efficiency of any response when a problem occurs. Understanding the parameters influencing production is a challenging task, especially if no straightforward link is apparent between the monitored parameters and occurring problems. The task becomes even harder if no trace of any actions or tuning of the system is left for analysis.

Concerning data availability, as often in the electronics industry, it was abundant but only little of it was relevant to the problem that had to be solved.

For all the previous reasons, the first step towards developing a control tool is to solve the traceability problem, not only in terms of data, but most of all in terms of knowledge from the production floor. Collection of this knowledge and analysis of the subsequent enriched information can greatly improve the ability to control the production. In such a case, a two-phase approach was considered and Fig. 6 describes the underlying idea.

– First, an expert knowledge collecting system (EKCS) was designed in order to increase the amount of information about the process and how to control it. Each time a problem appears on the production line and a machine has to be stopped, a new database entry is generated in order to record the main breakdown information. This information is in three categories: the time of the event will allow cross analysis with parameters from other databases, whereas the cause according to the expert and the adjustment needed for the corrective action represent the collected knowledge.
– Once the traceability problem has been solved, the resulting tagged data from defective situations is used for further automatic analysis. Trends can be analyzed and procedures for future corrective actions as well as strategies for defect prevention can be defined. Improved knowledge of manufacturing parameters and their effects will lead to improved process control and higher product quality.

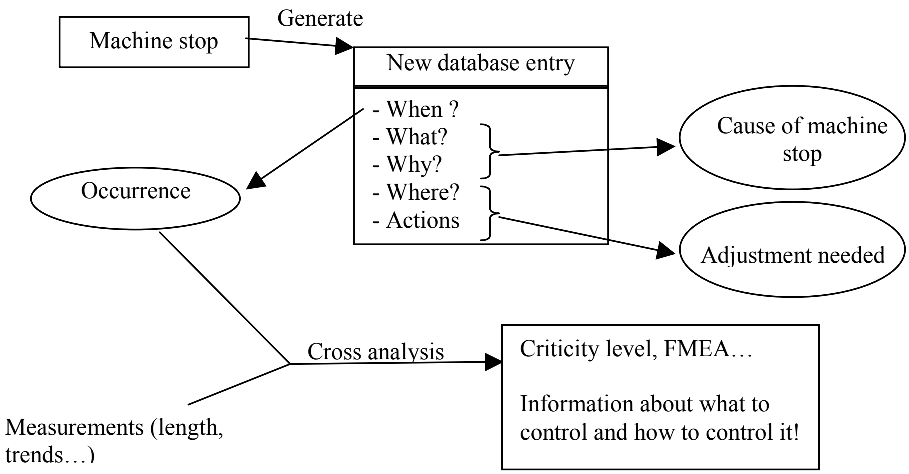


Fig. 6. Flowchart representing the Expert Knowledge Collecting System (Gebus *et al.*, 2004).

3.3.2 Development of a prototype version of EKCS

The development steps presented hereafter are a modification of the approach presented by McGraw (1994) that aims at integrating knowledge acquisition in the expert's work so that it becomes systematic without giving the impression of being compulsory. This can be achieved by adding automated knowledge extraction features to user-friendly interfaces used daily.

Identification and characterization of the users

Displaying information is not sufficient for a successful interface; the system operator must be able to input, receive and use the information. Therefore, any interface used for knowledge extraction should be tailored to its users. In the studied case, three types of users were identified:

- **Line operators:** Line operators are considered to be the main users of the interface and therefore the main knowledge provider. They are only input providers, meaning that they generate knowledge while using the interface, but they are usually not the ones for whom this knowledge is intended. They feed the system with the basic information about machine stoppages as well as the necessary corrective actions.
- **Team leaders:** Like line operators, they are input providers but can be considered as regular output users. The level of knowledge they possess about the process is higher than line operators and their comments richer in information. The task of the team leader is to supervise line operators and help them in solving unknown and therefore unregistered problems. In a way, line operators already use them as a knowledge pool

for richer information. As output users, team leaders make weekly quality reports for the line operators.

- **Quality engineers**: They are casual output users. This means that they are only expected to collect the knowledge from a database created for this purpose. The frequency at which they will use this database is independent from the process. Their interest is to get the knowledge in a way, which is most suitable for a comparative analysis with the existing measurement data. Their primary aim is to research the causes of recurrent problems, but they can also use the system to define new long-term strategies for production or maintenance.

According to this description, the line operators are the main users. This information is useful when choosing the style of the interface. In this case, a graphical interface using, for example, checkboxes might be a good choice since the user must be able to link an on-screen information with a real physical process. A text-based interface would be confusing and require more attention from the user.

Definition of the process model

After having identified the main users, the next step is to define the process model. The aim here is not to specify the process, which is something that should be done much earlier, but rather to characterize the interactions between the process and the different knowledge sources. In other words, it should be specified what data and information can be collected from the process (monitored information), and what can be acted on (actions taken on the process).

- **Monitored information:** In the case study, six parameters concerning the dimensions and specifications of manufactured products were monitored. In addition to data, visual supervision of the operator standing next to the production line also brings information. This information is much more difficult to model but can be ignored as measurements have a higher resolution than visual inspection. A problem serious enough to be visible cannot be missed by the monitoring system.
- **Actions taken on the process:** Process-monitoring produces input information whereas the response of the operators to some fault situations produces output information in the form of tuning of the manufacturing tools and repairing actions.

It has been noticed that most of the process monitoring is left to the operator's discretion. Therefore, a lot of research still needs to be done in standardizing these concepts.

Definition of a general fault model

The fault model is the result of the process model analysis. It analyzes the cause and effect relationships between ascending and descending information. Tools such as fault trees or fault mode and effect analysis (FMEA) are well suited for this kind of task even if they can only be detailed after cross analysis of the extracted knowledge with the measurement data as shown on Fig. 6. The aim is not to produce an exhaustive and fully detailed fault model, but only to give a general direction for the design of the interface. But there is no need in reinventing the wheel and if some fault modes are already known and identified, they should be integrated in the interface design. This will not only create a familiar framework for the users, but also simplify the knowledge extraction process and enable portability.

Design of a prototype interface

The main directions for the design of an interface for knowledge extraction that will be tested on the factory floor have been defined. The main specifications are:

- The formulated user types suggest that a graphical interface would allow a close connection and good integration with the production system.
- In order to generate a more detailed fault model, the parameters of the process model should be collected. A permanent record of all input information and all output actions must therefore be maintained and tracked by chronology and by incident number.
- Known faults should be included, but any other entries are also relevant, which suggests that this information should be recorded automatically in standard format.

Fig. 7 shows the choices that have been made for the prototype version of EKCS. When a machine stop occurs, operators are requested to push a button before inspecting the machine. This will automatically create an entry for date and time in a dedicated database, and wait for the operator to return with the relevant fault information. The following windows will then guide the operator through the knowledge acquisition process. This is done in two separate steps:

- The first step is to locate the fault on the production tool by providing the operator with pictures of the product and check boxes representing the different stations of the production line. This choice was made because the localization of the fault is a simple way to specify a problem and to narrow down the possible causes.
- The second step is to feed EKCS with information about the type of fault. The known faults are already listed, but since this list is not exhaustive, the operator can make additional comments.

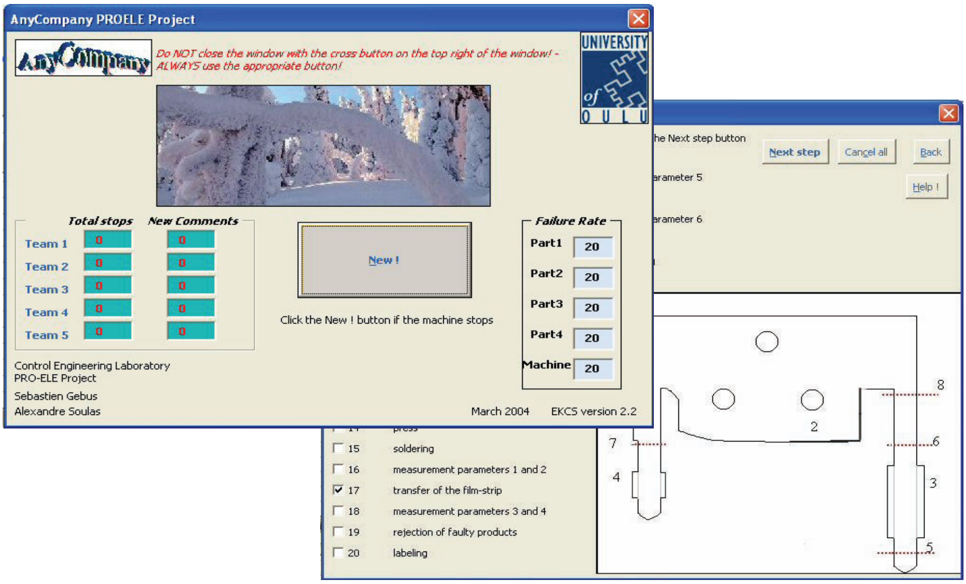


Fig. 7. A prototype version of EKCS (Gebus *et al.*, 2004).

Test, debugging, and redesign

It would be desirable to do it right the first time and some design methodologies claim to achieve just that, thus ignoring the test and redesign steps. Such an approach might be risky considering the characteristics of the tool under construction. Despite following the step-by-step methodologies given, it seems adventurous to claim to have created successfully at first draw a tool which is context-specific, user-specific, and problem-specific. This has been confirmed by problems encountered with the prototype version of EKCS.

Test conditions

EKCS was implemented on the factory floor and tested for a period of two weeks, during which all machine stops were systematically recorded, representing all cases together 183 machine stops. This was compared to the more than 300 000 lines of data collected over the same period of time for the measured production parameters. The information was analyzed and combined with the additional feedback provided by the operator about their experience with the interface. As expected, the very existence of this additional feedback suggests that at least some parts of the interface needed to be redesigned.

Four categories of problems with prototype version

Analysis of the data and feedback from operators made it possible to identify the following four categories of problems, each one leading to redesign propositions.

- **Misunderstanding of the task:** The first version of EKCS was confusing to the operators as far as the localization of the defects was concerned. This was noticed from the comments given by operators, but even more from the collected data. Instead of locating the occurrence of the fault, operators located its detection. This problem was solved by improving the design of the interface and providing clearer guidelines for users. One should also remember that any improvement was limited by the users' involvement and understanding of the purpose of their task.
- **Unexpected situations:** When designing an interface for knowledge extraction, it is mainly intended for some specific cases. Even for adaptive knowledge-based systems, the interface is usually the least flexible part of the system. It is therefore not surprising that unexpected situations cannot be dealt with. In the case study, this was illustrated by faults being linked to unforeseen problems with a non-functional part of the product that was not expected to have any effect on control parameters.
- **Expected unreferenced situations:** Different from the previous category, these situations represent the gaps in the initial fault model used during the design phase. Analysis of the data made it possible to reference new fault patterns resulting in machine stops.
- **Lack of motivation:** Test data consisted of nearly 300 000 lines of data, representing 183 machine stops, out of which only 70 were commented on. It was suggested to add features in the software aimed at motivating the users and ensure that it will be used more systematically even, for example, during night shifts.

Prototype versions, even if they might be time consuming, 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.

3.3.3 Final version

When developing a KBS, problems encountered during the prototype phase are seldom concept-related, but are rather linked to the effective implementation of such a tool and its ability to communicate within a factory environment. This is because users are the most unpredictable part of the design process. From their feedback, three different user-oriented axes of improvement have been identified, ***Usability***, ***Usefulness***, and ***Usage*** (3U).

Usability

In the case study, the information input needed to be much simpler and faster in an environment that can really be considered as intuitive for users. Only this way will it become a part of the operator's everyday work instead of an additional burden.

Furthermore, the time needed to update the system and adapt it to other production lines was too long, reducing its portability. Checkboxes were discarded as a possible solution. Adequate in very simple and most of all static cases, they become confusing when more options need to be added on the interface affect the appearance of the UI. 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.

A solution was eventually chosen to solve the portability problem while keeping a unique framework for the users. Instead of choosing among an increasing number of options, defect localization was done 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 while keeping the environment and the interface very familiar, resembling a factory floor.

Usefulness

The very first and most important target of any system is to be useful to the users. This includes the primary users, but also all the secondary users, whose tasks are among others to maintain and update the system. The case study in section 4 will show that a system aimed at simplifying operators' work, but updated by design engineers, does not have any long-term continuity. Even if very little additional work for design engineers led to a drastic decrease in the operator workload, the system was not properly maintained because the first group did not see any direct interest in performing the task.

The problem of motivating every single user of a system into using it was confirmed in the prototype version of EKCS. The solution retained was to transform the EKCS from a simple fault collecting system to a factory wide information sharing system providing different levels of added value depending on the users. 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.

Usage

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.

Another way to motivate operators to use EKCS is to include features aimed at monitoring the usage of the system. This can be done in a repressive way by allowing comparisons between different teams. It has been noticed that the usage of the tool can be less frequent during night shifts or by a certain group of workers. Operators can also be motivated in a more incentive way by providing them with information about how their work is benefiting improved production quality.

The new structure

The EKCS 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 an administrator. His role is to update the system according to the information collected from the production floor. Fig. 8 describes the information flows between the different entities and Fig. 9 to Fig. 11 provide further details about the new structure of EKCS and design choices that were made. The text in these screen captions, however, is of no interest here. As an information sharing system benefiting all the users, only closed loops of information flows should be present. Operators provide defect information that will be 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 for updating the system also uses defect information casually. New setups are stored in the database to be used by the two other interfaces. Since the quality department is the main beneficiary of an up-to-date system is the, it is recommended to assign this task to one of the quality engineers.

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.

From the practical point of view, the three interfaces use a unique database allowing anautomatic and immediate update of the system. This database is stored on a SQL server providing the needed flexibility that was missing in the previous 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 a certain production line. This was added in order to make the administrator interface a fully integrated subsystem.

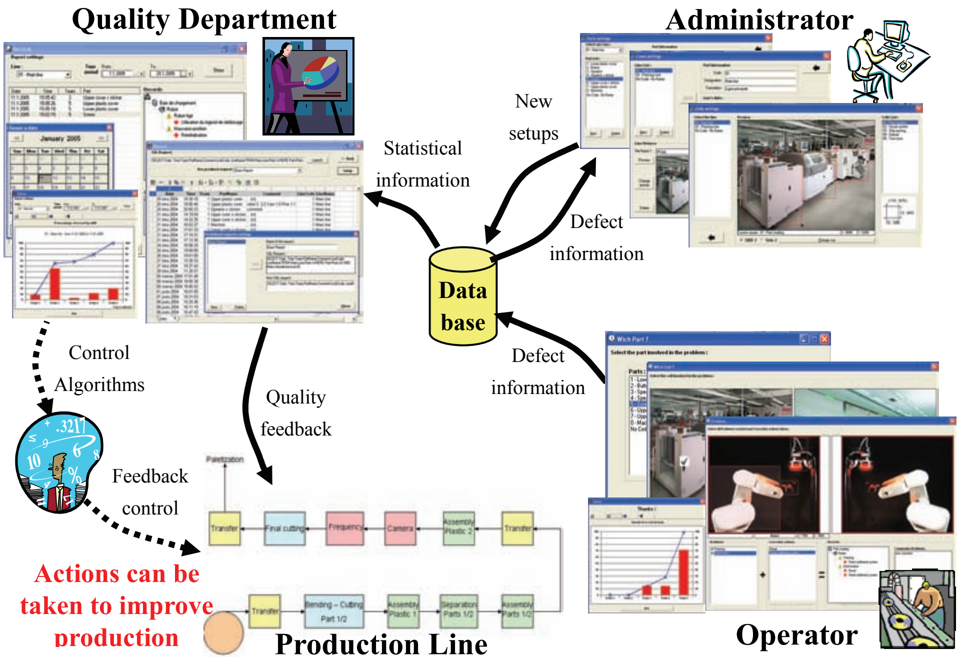


Fig. 8. General structure of EKCS with information flows between the different entities (Gebus *et al.*, 2006).

Overview of the different interfaces

Fig. 9 shows the structure of the administrator interface. The different steps to create or update line information are all accessible from the main window. The full creation procedure is done by selecting successively the different options provided in that interface.

- **Step 1**: Choose the localization for the new database or select an existing one.
- **Step 2**: Create or choose a production line and modify the information (*i.e.* the designation and up to two pictures best representing it).
- **Step 2'**: Do the same with the different product parts.
- **Step 3**: Create cells for the previous production line and modify the information.
- **Step 4**: Create problem groups for the cells. A problem group is an area on a cell, onto which has been attached a list of problems and associated corrective actions.
- **Step 5**: Define clickable areas for every cell on the production line pictures and in the same way for every problem group on the cell pictures.

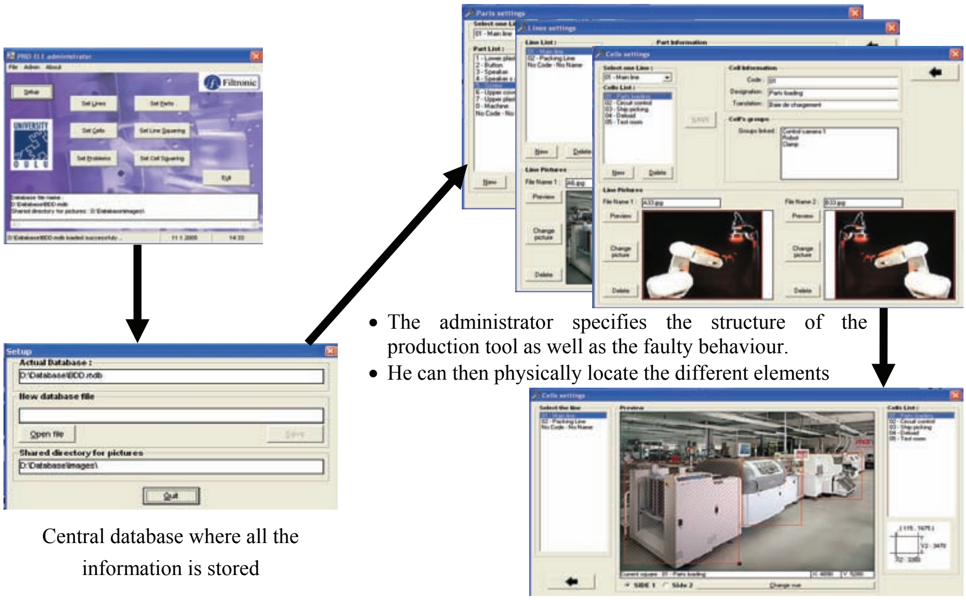


Fig. 9. Structure of the administrator interface (Gebus *et al.*, 2006).

Of the three interfaces, the one for the operators is the simplest and the most intuitive. The structure is given in Fig. 10. Setup options are available but since they consist of choosing the database location, selecting the current line, and downloading new pictures, this is done only once when the system is implemented on a new production line or when pictures have been updated by the administrator.

In normal use, data input is done by choosing the product part from a list, choosing the cell from the production line pictures and choosing the problem group from the cell pictures. At this point, the list of known causes and corrective actions is available as well as a comment window in case of unreferenced problems. Selection of causes and actions generates automatically a fault tree that is, in addition with charts representing short-term quality and usage information, accessed afterward during quality meetings.

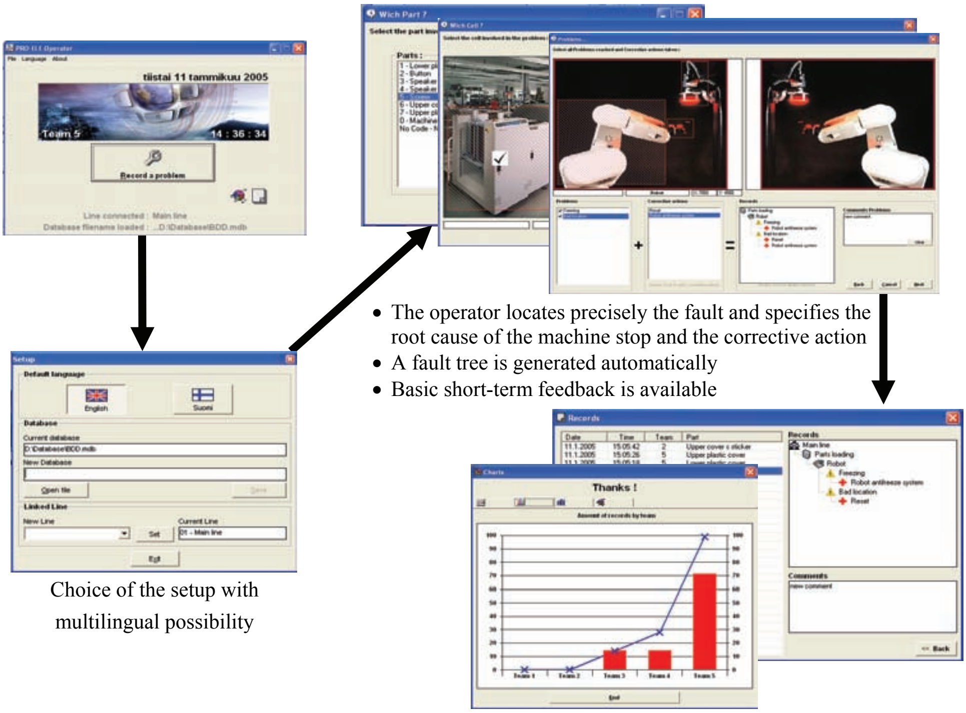
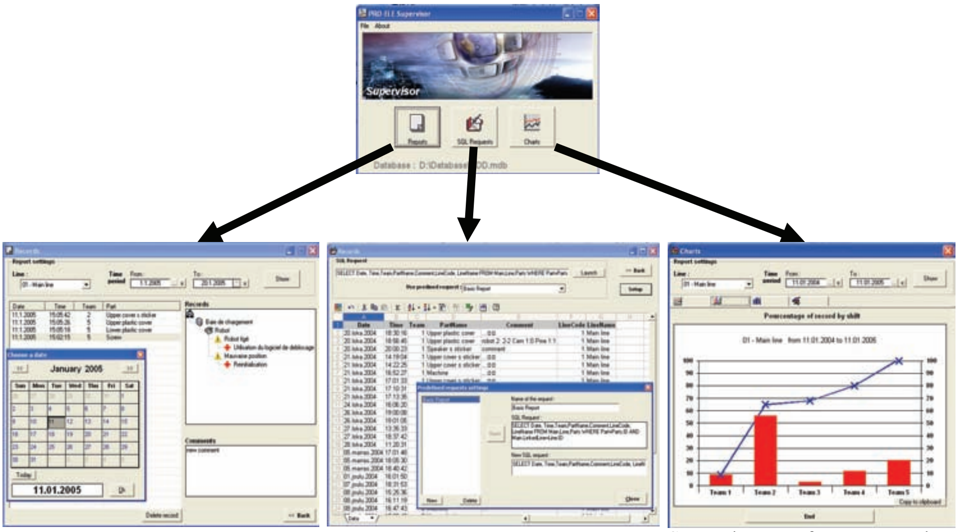


Fig. 10. Structure of the operator interface (Gebus *et al.*, 2006).

The supervisor interface is more complex than the previous ones and offers three main options as shown in Fig. 11. The same kinds of reports as well as the same charts as in the operator interface are also available here, but with no limitations in time. 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.



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

Fig. 11. Structure of the supervisor interface (Gebus *et al.*, 2006).

3.3.4 Results and further work

Through the effective implementation of EKCS on the factory floor, employee involvement towards quality issues has been promoted. Workers are now actively using the various features at different levels of the company. Line operators, for example, use the tool for their weekly quality meetings, thus encouraging its systematic daily use, whereas the quality department can benefit from its customizable report capability. It seems that for the first time since collaboration with the company in this case study started, cross analysis of data and expert knowledge will be possible. Hence, it is now possible to assess if the process is controllable with the current monitored parameters (product functional quality instead of process operational quality) or if in this case the entire monitoring strategy should be changed.

The supervisor interface is probably the most interesting of the three interfaces, not so much for its complexity but rather for the potential that it offers. It proves 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, fault trees etc.). This interface is therefore the backbone for the implementation of any future monitoring and control algorithm.

3.4 Conclusions

When building a KBS, knowledge acquisition is the most time-consuming part of the process and many problems may arise. Whether manual methods or automatic acquisition systems are used, the approach that is used in the end 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 for users with various degrees of knowledge. In addition to this, the complexity of any interface must be sufficient enough to grasp 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.

4 Defect localization on printed circuit boards

Production lines in electronics manufacturing are fairly standard in their design. They consist of paste printing, component placement and soldering. They are linear and sequential. It should therefore be quite easy to control the production between each sequence and prevent problems. In practice this is not done due to a scarceness of resources.

In practice also, and paradoxically, large amounts of these resources are allocated to repair-related activities. This was the case some decades ago in heavy manufacturing such as the automotive industry where rework operations constituted not only a factory within a factory, but also the most labor- and equipment-consuming operation. This is still the case today in the electronics industry.

Furthermore, the increasing complexity of electronic products has lead to an explosion of material costs. Detecting defects and repairing Printed Circuit Boards is therefore not a choice, but has become a necessity that calls for the development of a new fault diagnosis system.

4.1 Diagnosis in electronics

The complexity of modern systems places new demands on system maintenance. Every system has a mission to perform. When it fails, the job of maintenance is to diagnose and repair the system as rapidly as possible so that it can return to correct operation.

The general problem of diagnosis is extremely difficult. Systematic identification of malfunctions by means of symptoms can involve complex test sequences to ensure complete fault coverage, as explained by Simpson & Sheppard (1994). Optimizing such a diagnostic sequence of tests is known to be an NP-complete problem.

Integrated diagnosis is an attempt to solve this problem. Keiner (1990) defines integrated diagnosis as “a structured process which maximizes the effectiveness of diagnostics by integrating the individual diagnostic elements of testability, automatic testing, manual testing, training, maintenance aiding and technical information”. This definition is interesting because it encompasses different methods and goals. The primary goal is to optimize field maintenance. It includes minimizing the mean time to isolate system faults and the mean time to repair, but also training considerations. By reducing the need for specialized diagnostic skills, it is possible to reduce the impact of losing experts and to minimize training costs. More generally, three different aspects of diagnosis can be identified:

- ***Detection***, which refers to the ability of a diagnostic strategy to identify that a failure in some system has occurred;
- ***Localization***, which is the ability to say that a fault has been restricted to some subset of possible causes; and
- ***Isolation***, which is the identification of a specific fault through some test or diagnostic strategy.

The method proposed and illustrated by the case study in section 4.4 uses an approach that tries to solve several of these problems. Furthermore, as fault recovery is often not given enough importance in diagnostic systems, it shows how proper use of diagnosis information can lead to efficient repair procedures.

4.1.1 Specific needs and the zero defects utopia

Failure analysis in the electronics industry provides many challenges, which fall mainly into two categories: failure site isolation driven by device complexity and reduced accessibility, and physical analysis driven by a smaller size of device features and new materials. The second category, however, is out of the scope of this thesis.

In the context of increasing complexity, Wagner (2001) describes the following challenges of failure site isolation:

- ***Localization and electrical characterization***, which is the iterative process of reducing the scope of a failure analysis problem. The initial step toward localization of a faulty circuit element may be to reduce the scope from the entire circuit to a circuit block.
- ***System-on-a-chip***, which is the evolution of integrated circuitry. The biggest issue is associated with the complexity of the chip and testing is equivalent to bringing the device into a failing electrical state.
- ***Detection and characterization of non-visual defects*** such as short-circuits or inversion of charges within components will be an increasing problem.
- ***Verification and testing strategies***, as an NP-complete problem, will see its complexity increase exponentially with the complexity of the underlying circuit. Complete test coverage, as a result, becomes more and more improbable.
- ***Fault isolation and simulation software*** has to assist the work of test designers more and more.
- ***Cost of failure analysis*** has risen steadily over the last decade with the increase in the complexity and volume of data.

Although it covers many of the challenges associated with failure site isolation, this list may focus too much on the testability and test designers' point of view at the expense of the consequent repair and rework activities.

With the ever-increasing complexity of the devices, defects are very likely to occur, no matter how much attention is paid to their prevention. This can be shown by very simple facts.

1. For the final consumer, an electronic device can usually have only two very distinct states: “Working” or “Not working”.
2. The “Working” state depends on the electronic components used to make the device, all of which have a specific function.
3. As a consequence, the “Working” state depends on every single component and even the smallest defect in one of them will have a significant impact on the overall performance of the device. In other words, electronic manufacturing is an environment where zero defects are a necessity.

From a purely statistical point of view, however, production with zero defects is an impossible target, and no production system can or ever will achieve it, as by increasing the amount of components on a PCB, the defect rate is also more than likely to increase drastically. This can be shown by the following example: If X is the yield for individual components and Y the amount of components, then the overall yield for the PCB will be X^Y. For example, a PCB made out of 1000 components, with a yield of 0.999 for each one of them, will have an overall yield of 0.999^{1000}=37%. It means that two thirds of the products would need some rework.

Modern production systems for components are characterized by ppm (parts per million) levels of defects, thus allowing an increase of yields for individual components. These, however, are independent of the performance of any electronic assembly process and, unfortunately, modern devices are also characterized by millions of elements. As a result, even if it would be better to tackle problems at their sources, predictive and preventive actions on the assembly line do at some point reach their limits. For this reason, attention should be given to diagnosis and repair activities linked to unavoidable end of line defects.

4.1.2 Defect localization problem

Balakrishnan & Semmelbauer (1999) define electronic diagnosis as “the process of identifying the components or connections that are responsible for the malfunction of a defective printed circuit board so that corrective action can be taken both to repair the board and to improve the process”. One of the aims of diagnosis is therefore to improve electronic maintenance by permitting a technician prompt and easy access to the information needed to identify the fault and make the repair. However, since electronic components cannot be repaired and their individual cost is negligible, the understanding of a malfunction is not considered a major concern. Therefore, the most important goal for a diagnosis system from the repair and rework point of view is to provide information about the location of the faulty component.

Furthermore, since repair is a post-production activity, the problem of fault localization is often approached from the point of view of the final testing. At this stage, in-circuit tests become more and more complicated to implement, and most electronic devices are going through functional testing. This is already a common procedure in fault detection at the chip level as explained by Stout *et al.* (1998). However, unlike structural testing methods, functional testing does not allow direct localization of a fault. New methodologies are therefore necessary to analyze data from functional testing and locate faults. Furthermore, in order to facilitate repair operations, any new methodology would also have to be incorporated into a decision support system aimed at repair workers.

4.2 Different approaches

The increasing complexity of electronic systems will soon exceed their ability to be tested and maintained unless improvements are made in testability, diagnostics, and test programs. In attempting to address diagnosis problems, several tools have been developed to build efficient fault-isolation strategies using many different approaches.

Some methods permit blind data analysis based on the fact that faults normally change the correlation of process variables. Yang *et al.* (2002) propose a method based on principal component analysis (PCA) able to extract from data direct information about failure causes. PCA is one of many multivariate regression methods described by Hyötyniemi (2001). These approaches, however, are out of the scope of this thesis as it focuses on knowledge-based approaches.

Besides brute force, which is no longer feasible past a certain degree of complexity; fault detection and diagnosis methods include the use of neural networks (El-Sayed & Alfuhaid, 2000), fuzzy logic (Ulieru, 1994), rule-based systems (Winter, 1992), model-based reasoning (McKeon & Wakeling, 1989), and case-based reasoning (Balakrishnan & Semmelbauer, 1999). These various approaches often present weaknesses such as the inability to deal with incomplete data or knowledge. However, because the weaknesses of different methods do not necessarily overlap, the idea to create hybrid methods emerged over a decade ago (Isermann & Ulieru, 1993). Among the extensive opportunities for combination, in the context of fault diagnosis one could cite neuro-fuzzy approaches (Sutton, 1992), fuzzy expert systems (El-Shal & Morris, 2000), or even CBR-based hybrid systems (Wang & Wang, 2005) (Kuo *et al.*, 2005)

Combining methodologies can improve system efficiency, but today's processes are not only complex, but also non-linear and multivariable, thus some limitations are likely to remain. New approaches are therefore necessary that take into account interactions between the variables, whether these are quantitative or derive from expertise. Better information and data models can trigger the development of a new generation of smart adaptive systems that will be able to handle both uncertainty and complexity.

4.3 Linguistic equation based fault detection

The linguistic equations (LE) approach developed by Esko Juuso at the Control Engineering Laboratory of the University of Oulu (Finland) is a possible solution to some of the limitations found in hybrid methods. While originating from fuzzy sets systems, the difference lies in the fact that membership functions are replaced by membership definitions that operate like smart non-linear scaling functions. Through this scaling, data is transformed for use in a linear system. In effect, these linear equations advantageously replace fuzzy rules as they can potentially compact a large rule-base into a few linear equations.

The method has already been adapted for diagnosis problems. Juuso (1994) for example, presents theoretically what a fault diagnosis system based on the LE framework could be. Later, the method was also used by Komulainen *et al.* (1997) to develop an approach applied to functional testing based on fuzzy rules and linguistic equations, and by Juuso *et al.* (1998) in a case-based reasoning type approach.

The following sections are an attempt to introduce and explain in broad terms the concept of LE through examples and analogies with the underlying fuzzy sets theory. More details can be found in Juuso (1999) and Juuso (2004). The author's own contribution is then presented through case study No 2, where an LE approach is associated with design knowledge for the first time to solve a fault localization problem.

4.3.1 Linguistic values

The general approach used in LE is rather similar to the one used in fuzzy logic. Membership functions are replaced by scaling functions also called membership definitions. These membership definitions provide non-linear mappings from real data, defined with feasible ranges, to the linguistic values from the real-valued interval [a b]. Hence fuzzification and defuzzification are the result of these scaling functions. Similarities between the two approaches go one step further since membership functions can be generated from membership definitions, feasible ranges being used as an intermediary representation.

A feasible range is defined as a fuzzy number characterized by a trapezoidal membership function as shown in Fig. 12. This membership function is known as the operation area and includes two parts. The main area of operation, also called the core area, represents the interval in which a variable can vary under normal system conditions. The second part is the underlying support area, which represents the whole range of variation of that variable. In theory, the support area is defined by the minimum and maximum values of the variable. This means that any value x_j of a variable j should always verify the condition:

$$\min(x_j) \leq x_j \leq \max(x_j) \qquad (1)$$

The limits of the core area, $(c_l)_j$ and $(c_h)_j$, can be defined on the basis of expert knowledge, in which case they are fixed, or extracted from data. The data-driven approach can, however, face two main problems. First, as we will see in the case study, this might result in [a b] being slightly different to [min(x_j) max(x_j)]. Secondly, calculation of the limits is very sensitive to outliers (*i.e.* values far from most others in a set of data), especially if the core area is very narrow compared to the support area.

In addition to the previous four points delimiting the areas, a centre value c_j can be defined for each variable j, dividing in effect the whole support area into a lower and an upper part. This value can be defined as the mean or median of the corresponding data set, in which case c_j not only divides the support area into two parts but also does the same with the core area.

Feasible ranges can also be seen as a bridge between membership functions and membership definitions. Since they should be consistent with membership definitions, they are defined together. However, they can also be generated from membership functions when existing rule-based fuzzy systems are converted to linguistic equations. Fig. 12 shows the analogies between the different representations. Although only five membership functions are used in Fig. 12, in theory any number of membership functions could be defined over that interval. The shape of the membership definition will encompass the information about the underlying membership functions (Juuso, 2004).

In theory, [-p p] could be used for the entire linguistic interval with [-p/2 p/2] being the linguistic interval for the core area. However, for convenience, [-2 2] is often used as it describes more easily the conventional scaling [negative big, negative small, neutral, positive small, positive big]. Furthermore, the idea of having a simple interval, [-1 1], in which values vary under normal system conditions is handy in the context of fault diagnostics. In the rest of this chapter we will therefore use [-2 2] as the linguistic interval.

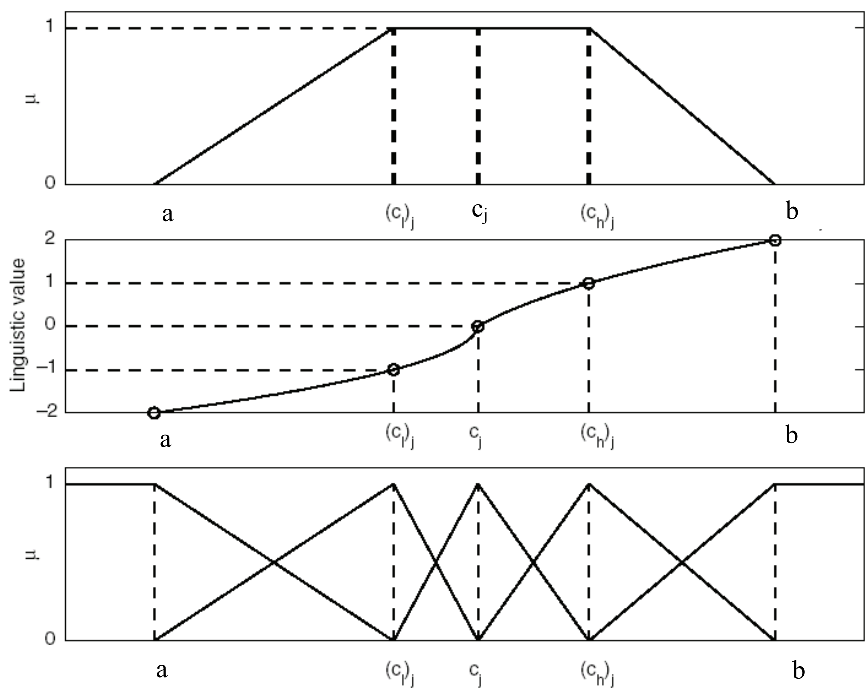


Fig. 12. Feasible range, membership definition, and membership functions (Juuso, 2004).

Membership definitions are used to provide non-linear mappings from the operation area to a linguistic range, defined by the interval [-2 2]. However, they are defined in the opposite direction by the functions:

$$\begin{aligned} &\left[-2\ 2\right] \to \Re \\ &X_j \mapsto x_j \end{aligned} \tag{2}$$

where x_j is the value of variable j and X_j is the corresponding linguistic value.

Membership definitions can be generated automatically from the training data by fitting a second order polynomial against the data points. Thus membership definitions consist of two second-order polynomials: one for negative values of X_j and one for positive values. Equation (2) can therefore also be written as follows:

$$\begin{aligned} &\left[-2\ 0\right] \to \left[\min(x_j)\ c_j\right] \\ &X_j \mapsto f_j^-(X_j) \end{aligned} \tag{3}$$

and

$$\begin{aligned} &\left[0\ 2\right] \to \left[c_j\ \max(x_j)\right] \\ &X_j \mapsto f_j^+(X_j) \end{aligned} \tag{4}$$

These two functions must be strictly monotonic and increasing in order to have a reversible relationship between the real value of a variable and its linguistic value. Furthermore, the functions must overlap at the linguistic value 0. This can be achieved by using the feasible range of a variable as a starting point in the construction of membership definitions. $\min(x_j)$ and $\max(x_j)$ can be associated to the linguistic values -2 and 2, while $(c_l)_j$ and $(c_h)_j$ can be associated with the linguistic values –1 and 1, whether these are extracted from data or defined by expert knowledge. The linguistic value 0 is associated with the centre value c_j, also defined by expert knowledge or by defuzzifying with the centre of gravity method. For automatic generation of membership definitions, the mean or median of the data are used directly as the center value.

After having defined the five characteristic values of the membership definitions

$$(\min(x_j),-2),((c_l)_j,-1),(c_j,0),((c_h)_j,1),(\max(x_j),2)$$

the next step is to calculate the coefficients of the second-order polynomials f_j^- and f_j^+. Since these overlap at the value c_j, they can be defined by:

$$\begin{aligned} f_j^- &= a_j^- X_j^2 + b_j^- X_j + c_{j,} \qquad X_j \in \left[-2\ 0\right] \\ f_j^+ &= a_j^+ X_j^2 + b_j^+ X_j + c_{j,} \qquad X_j \in \left[0\ 2\right] \end{aligned} \tag{5}$$

with $a_j^-, b_j^-, a_j^+, b_j^+$, the coefficients of the polynomials f_j^- and f_j^+.

As the center point c_j has already been defined, calculating the other coefficients is equivalent to solving the linear system:

$$\begin{aligned} 4a_j^- - 2b_j^- + c_j &= \min(x_j) \\ a_j^- - b_j^- + c_j &= (c_l)_j \\ a_j^+ + b_j^+ + c_j &= (c_h)_j \\ 4a_j^+ + 2b_j^- + c_j &= \max(x_j) \end{aligned} \tag{6}$$

In order to respect strict monotony, it is also necessary to verify that the derivatives of the resulting functions remain positive over their respective intervals. If this is not the case then the coefficients of the polynomials have to be adapted, and [min(x_j) max(x_j)] is replaced by [a b], resulting in a slight expansion or contraction of the associated feasible range.

When using a linguistic equations approach, non-linear scaling is performed twice: once from real values to the interval [-2 2] and a second time from the interval [-2 2] to real values.

In the case of polynomial membership definitions, scaling from real values to the interval [-2 2] is accomplished by inverting equations (5) and applying the result within the support area, whereas outliers are attributed the linguistic values –2 and 2. This results in the following equations:

$$X_j = \begin{cases} 2 & \\ \dfrac{-b_j^+ + \sqrt{b_j^{+2} - 4a_j^+(c_j - x_j)}}{2a_j^+} & \\ \dfrac{-b_j^- + \sqrt{b_j^{-2} - 4a_j^-(c_j - x_j)}}{2a_j^-} & \\ -2 & \end{cases} \text{with} \quad \begin{aligned} & x_j \geq b \\ & c_j \leq x_j \leq b \\ & a \leq x_j \leq c_j \\ & x_j \leq a \end{aligned} \tag{7}$$

Scaling from interval [-2 2] to real values is more straightforward since equation (5) can be applied to the output linguistic values to obtain the corresponding real output value of the linguistic system:

$$x_{output} = \begin{cases} f_{output}^-(X_{output}) \\ f_{output}^+(X_{output}) \end{cases} \text{with} \quad \begin{aligned} & X_{output} < 0 \\ & X_{output} \geq 0 \end{aligned} \tag{8}$$

4.3.2 Rule base

By using the procedures described previously to define the characteristic values of the membership definitions, these will more or less describe the distribution of a variable value over its range. Therefore, fuzzifying a variable through membership definitions and forming a linear system with those linguistic variables aims at removing non-linearity.

By continuing the analogy with fuzzy sets theory, linguistic equations are equivalent to the fuzzy rule base. With the LE approach, the linguistic model uses Linguistic Variables in order to generate equations and each of these equations can therefore be assimilated to a fuzzy rule. The basic form of a linguistic equation model is:

$$\sum_{j=1}^{m} A_{i,j} X_j + B_i = 0 \tag{9}$$

where X_j is a linguistic value for the variable j obtained from scaling and B_i is a bias term introduced for fault diagnosis. In this linear system, each equation represents a multivariable interaction. The direction and strength of the interactions between the linguistic variables are represented by the interaction coefficients A_{ij}. These coefficients can be represented as an interaction matrix A and a bias vector B, and the equation (9) becomes:

$$AX + B = 0 \tag{10}$$

In small systems, the interactions between the variables are usually quite clear. For more complex systems, however, selection of variables becomes necessary to retain only the combinations of variables with the strongest interactions. Different methods can be used to choose alternatives for variable combinations. The choice can also be improved on the basis of expert knowledge but practical cases have shown that in general the amount of variables per equation does not need to be greater than three to five. Larger groups of variables often lead to redundancies, thus coefficients close to zero or variables that can be entirely expressed by a combination of other variables of that tested group. Since only the variables with non-zero coefficients belong to the interactions, very large systems can be packed efficiently.

A tuning algorithm tries to reduce the error between the model and the data. The LE approach by its nature allows such tuning abilities, whether these are aimed at modifying the membership definitions or the linguistic equations.

Since only five parameters are needed for each variable, some of which can be extracted automatically from data, the LE system can easily be adapted, even on-line, by recalculating some of the five characteristic values and thereby changing the membership definitions when new data becomes available.

When tuning aims at modifying the linguistic equations, techniques based on neural networks or evolutionary algorithms as presented by He *et al.* (2000) can be used to recalculate the weighting of the interaction coefficients.

4.3.3 Application to fault diagnosis

Linguistic equations provide a linear approach to system analysis. Therefore the models that are generated can be completely inverted and used for example without an output variable. This property also makes the LE approach a suitable candidate for fault diagnosis, which aims essentially at analyzing the interactions between symptoms. This property is best explained through an example.

In a very simple two-variable case, there are only two variables X and Y and it is assumed that some sort of interaction exists between them. In this set of data it is also supposed that two defect situations can be identified, as shown in Fig. 13. In such a case, a fuzzy logic based approach would imply the application of membership functions to the original variables and sorting them into different categories. Fault detection would then be the result of fuzzy rules such as:

- IF x is "normal", THEN y is normal (no fault detected)
- IF X is "low" or "high", THEN Y is low (fault detected)

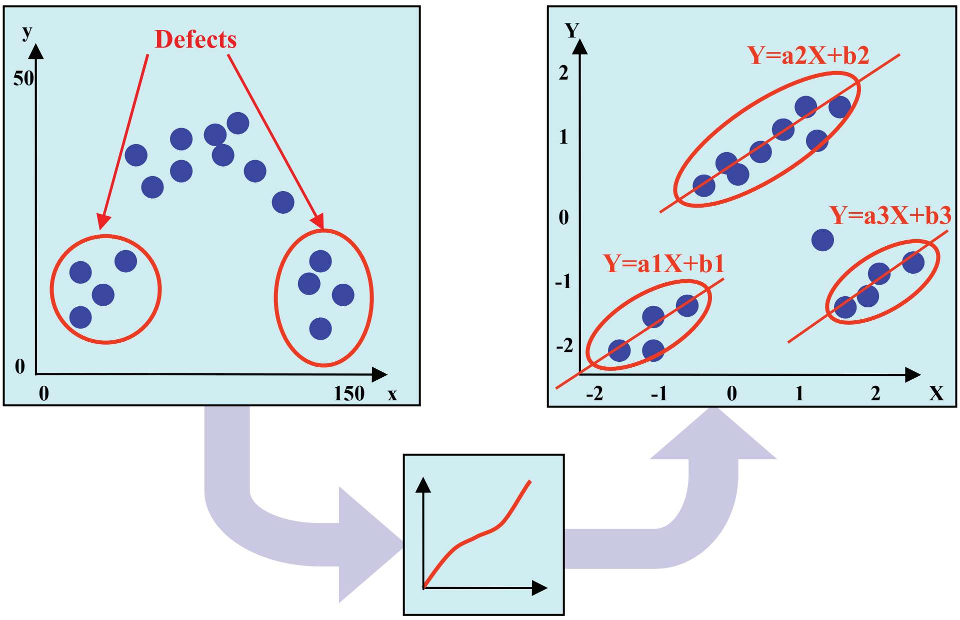


Fig. 13. Fault diagnosis concept with linguistic equations approach. After scaling, defects are identified by linear interactions among linguistic variables (Gebus & Juuso, 2002b).

By comparison, Fig. 13 illustrates how the LE approach would deal with this data. First membership definitions are applied to each variable, with the effect of scaling and linearizing the problem. Then, the linguistic model looks for linear interactions between the resulting linguistic variables. By applying clustering methods and linear regression to this linguistic data, three equations are obtained. Each of these equations can be represented in the form of a line in the two-dimensional space of the variables. In effect, instead of applying fuzzy rules, we now have to check how close the linguistic points are to each of these lines. Therefore, the previous fuzzy rules become:

– IF $Y - a_2 X - b_2 = 0$, THEN the situation is normal
– IF $Y - a_1 X - b_1 = 0$ or $Y - a_3 X - b_3 = 0$, THEN there is a defect

The previous case can easily be generalized to a larger number of variables. In theory, the only difference is the use of multi-linear regression instead of standard linear regression. Linguistic equations can then be represented by p-dimensional surfaces in an n-dimensional space with p $\in$ [1 n-1]. If p=1, then the two remaining variables are interchangeable. In the same way, if p=n-1, then the linguistic equation is a hyperplane of the space of the variables and there is no redundancy among them. In this case, the resulting linguistic equations are similar to equation (9) and the linguistic rule base can be derived from the interaction matrix and bias vector as follows:

$$\sum A_{0,j} X_j + B_0 = 0 \Rightarrow OK$$

$$\sum A_{i,j} X_j + B_i = 0 \Rightarrow Defect_i \text{ when i} \neq 0$$

In practice, especially if there is a large amount of variables, many of the coefficients of the interaction matrix are expected to be negligible and are therefore assimilated to zero. One reason for this is that not all variables are relevant to every defect, and unless in-depth knowledge about defects or extensive fault models are available, many variables are likely to have only minor weight in most linguistic equations. Furthermore, if too many variables are included simultaneously in an equation, redundancy is likely to appear and the problem will be over-fitted.

Each equation can be considered as a case-based model in the Case-Based Reasoning approach presented in Juuso *et al.* (1998). The decision is based on how much the left hand side of the equation differs from zero when the scaled measurement values are used, *i.e.* how close to the surface the measurement point is. The difference defines the degree of membership for the case. More generally, each case can also have more than one equation model, *i.e.* all the equations which should be true at the same time. Then the conclusions must be aggregated from the degrees of membership calculated for individual equation models.

4.4 Case study No. 2: defect localization on PCBs based on functional testing and expert knowledge

4.4.1 Context and constraints

The case study has been taken from a project realized during 2001 with a company producing electronic control equipment; mainly for the automotive industry. The goal of the project was to develop and implement a functional testing tool to be used on production lines as a decision support tool. Post-production testing meant that the aim of that decision support tool was to facilitate repair operations.

In the beginning of the project, operators had only their own personal experience when having to repair a PCB. This experience was not available in a way that could benefit new employees, nor was it of any use when new products were introduced. The only remaining solution for them was to search in the hundreds of pages containing the electronic schematics of the new product, thus gradually learning by doing in a boring and most of all inefficient way, especially considering the short life cycle of the products.

This inefficiency was even more regrettable considering that the design of the functional testing was done locally, thus knowledge availability was not a problem in this case. Quality control, however, fitted extremely well to the problem described in section 2.2 and illustrated by Fig. 14. When a given PCB goes through functional testing, measured parameters and pass/fail test results is stored in a test database. Later, when a PCB is to be repaired, fault information is stored to a quality database allowing the quality department to report yield information. Unfortunately, between these steps traceability of the PCB is lost and with it the possibility to have usable data.

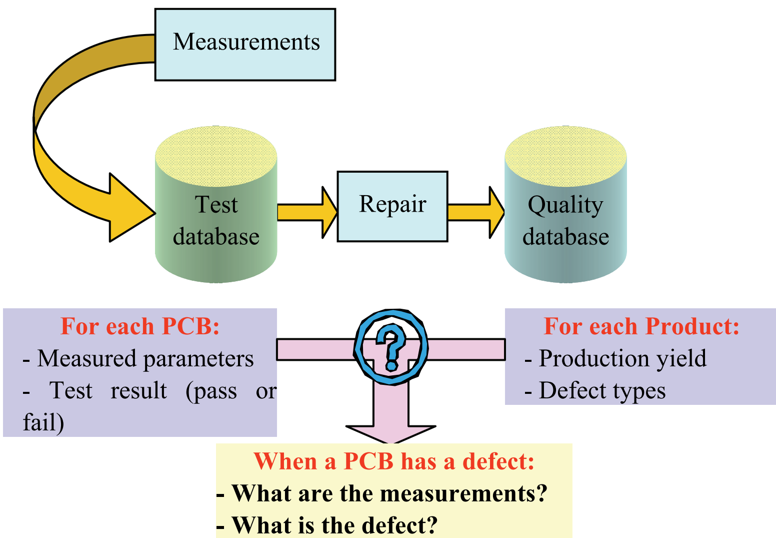


Fig. 14. Data and information availability problem in case study No. 2.

4.4.2 Solutions retained

Software development and implementation options must be selected according to some basic requirements:

- **Defect localization:** This has been defined as the primary objective since it is the one and only important information needed by the end-user.
- **Simple to use:** The end-user of the system is a typical repair operator. Therefore, information must therefore be tailored to him.
- **Robust:** The methodology used for defect localization must be efficient even under evolving conditions and new product introductions (quick ramp-up).
- **Self-tuning:** Knowledge maintenance is seen as an additional issue more or less related to the concern on robustness.

In order to achieve these targets, the tool has to be integrated in a wider DSS structure and make use of existing knowledge sources and other company systems already available. As the linguistic equations approach is an efficient integrating methodology for combining expertise and data-based modeling, and there is a long experience in using fuzzy logic in fault diagnosis, the fuzzy expert systems and their LE extensions were chosen to be the methodological basis of this project. However, general methodology had to be adapted to the specific data availability problem.

From a practical point of view, the Internet and intranet applications using web browsers were chosen as a solution for creating the required interfaces for the decision support system. This choice was made because the technology was already in use in some of the existing systems.

4.4.3 Modifications for the case study

The specific context of the case study is slightly different from the general case presented in section 4.3. Originally, a data-based approach had been considered to generate membership definitions and linguistic equations. However, the number of faults was too low for efficient data-based analysis; also the fault information as explained earlier was impossible to connect to the test measurements. Therefore, several modifications have been made to the general LE approach to integrate test design information as expert knowledge provided by test designers is the only reliable resource for building the LE system. The procedure described here has already been documented in Gebus & Juuso (2002a, 2002b, and 2003).

As shown earlier, LE-based fault diagnosis aims at analyzing interactions between symptoms and therefore no output variables are necessary in any equation. The problem consists of verifying how close a point is to any of the surfaces representing the linguistic equations. When a sufficient degree of membership is found, it is in effect the whole equation which is linked to an output of an LE-based inference engine. In the case study, inputs are therefore measured parameters from functional testing and equations are developed for each geographical area on a PCB where the fault may lie. Equation (9) becomes:

$$\sum_{j=1}^{m} A_{i,j} X_j + B_i = 0 \text{ with } X_j \text{ only inputs} \tag{11}$$

Furthermore, for each area, input variables are known to be independent because whatever independent fault is occurring, it will have an equivalent effect on the status of the area, which will be considered as defective. As a result, the bias vector disappears and the weighting coefficients are limited to the values 0 and 1. Equation (11) is therefore simplified as follows:

$$\sum_{j=1}^{m} A_{i,j} X_j = 0 \text{ with } A_{i,j} = 1 \text{ if } X_j \text{ is relevant to the area } i\text{, and 0 otherwise} \tag{12}$$

For each variable, functional tests are designed in such a way that an area is considered as defective if the measured variable crosses a certain control limit. Since this is a common rule to all variables and these control limits are known and well-defined, it makes sense to assign -1 and +1 as linguistic values to those limits. As for the central point, it is chosen for each variable as the linguistic value equivalent to the measured average of that variable during the system's steady state. It therefore represents the linguistic value that should be obtained under normal system conditions.

By making these choices, three characteristic values out of five are fixed, leaving only the support area of the feasible range to define using the general procedure described by Juuso (2004).

Fig. 15 shows a membership definition obtained with these choices. The shape of a membership definition can give a lot of information about the variable. This case, for example, provides the following information about that variable:

- Measurements are shifted towards the lower control limit.
- A "low voltage" will have a much bigger effect on the linguistic variable than a "high voltage" and will therefore be detected much faster.

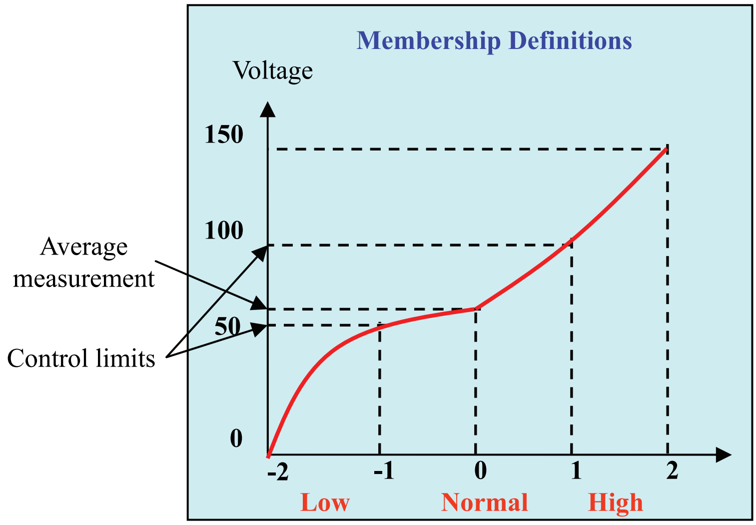


Fig. 15. Membership definitions when values of the variable are not centred (Gebus & Juuso, 2002b).

For each geographical area on a PCB, it is possible to state two general rules:

1. If every variable is "close" to zero, then equation (12) is "close" to zero and the area is OK.
2. Whatever the variable, it should never verify the condition $|X_i| > 1$.

By applying these two rules and including the deviation on the variables, equation (12) becomes the inequality:

$$|X_1| + \ldots + |X_n| < 1 + \sigma \text{ with } \sigma \text{ the deviation} \tag{13}$$

that has to be verified in order to have a defect-free area. The deviation has been calculated by:

$$\sigma = k\left|\sum \sigma_i - \overline{\sigma}\right| \text{ with } \overline{\sigma} = \sum \frac{\sigma_i}{n} \tag{14}$$

where σ_i is the standard deviation for the linguistic variable X_i and $\overline{\sigma}$ the average standard deviation for all the linguistic variables. Coefficient *k* is a coefficient that depends mainly on the amount of available data and the spread of the variables. A good case would be to have a lot of data for many variables with a similar standard deviation. In all other cases, coefficient k can be used to tune the system if it generates, for example, false alarms. The final form for the inequality is then:

$$\left|X_1\right| + \ldots + \left|X_n\right| < 1 + k\frac{n-1}{n}\sum\sigma_i \tag{15}$$

The linguistic model can be tuned using the general tuning methods when additional data becomes available. The easiest form of tuning can be achieved by modifying membership definitions. In this case, more recent data is used to recalculate the centre point corresponding to the value 0 as well as the limits of the core area corresponding to -1 and 1. Hence, the membership definitions are fitted once again.

Tuning of the equations is not considered because of the lack of defect information available. However, if in the future such information becomes available, one could imagine applying the general methodology for calculating coefficient weights based on real process data instead of being limited to expert knowledge.

4.4.4 Implementation of the LE-based decision support tool

In order to integrate the linguistic rule-based model within a wider structure of a decision support system, one solution is to analyze the functional interactions between the different parts of such a system and identify the required underlying technologies and shared information. Fig. 16 shows the second layer of a functional analysis of the desired system using an SADT approach. Zone 2 represents the function fulfilled by the linguistic model. It is the intelligent part of the system, whereas Zone 1 regroups functions aimed at collecting data and knowledge. Finally, Zone 3 is the feedback function that will be used to develop the end-user interface. With this approach, the initial problem stated as “detect and locate defects for repair operations” is now clustered into three smaller problems:

- Collect the required data;
- Transform it into information;
- Provide understandable and efficient feedback to the operator..

Since the second problem is taken care of by the linguistic model as shown before, only the data collection and the feedback problem remain after it.

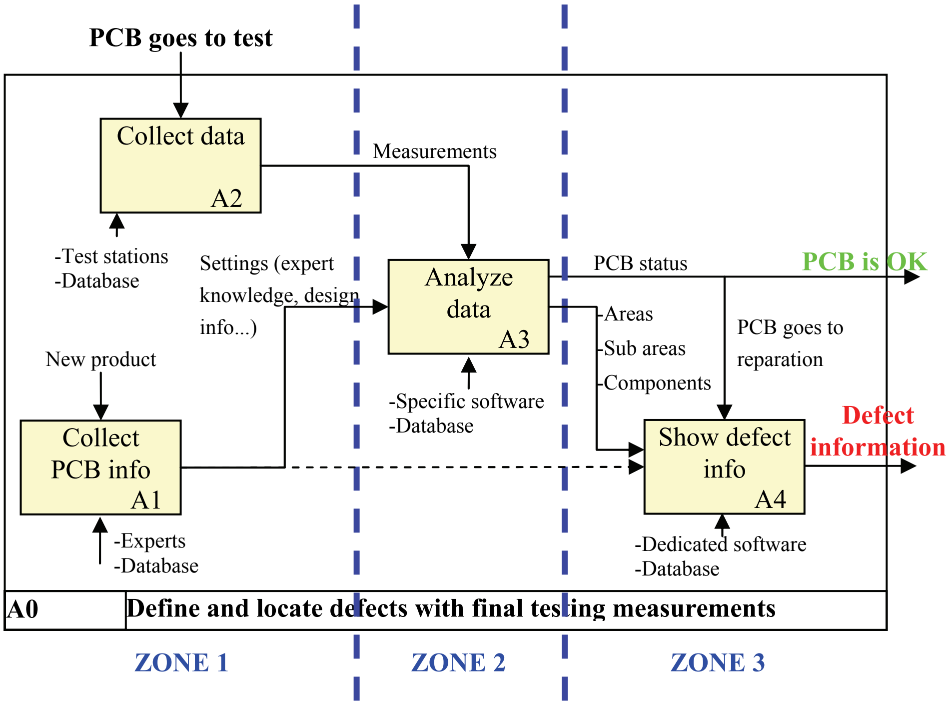


Fig. 16. Global functional definition of the DSS using an SADT specification approach (Gebus & Juuso, 2002b).

From Fig. 16, it can be noticed that the need for data collection can be divided into two parts. First, the data analysis sub-system needs a few measurements to define a centre to the feasible ranges. Then it needs design knowledge to define the limits to the core areas as well as functional localizations on the PCBs.

Most of the information required by the system can be retrieved automatically from existing databases (*e.g.* Pictures of PCBs, position of components…). However, designers need to feed the system with all the information necessary for the algorithm. The knowledge acquisition software has been developed using DHTML and Visual Basic in order to provide a user-friendly and already known framework for users. Furthermore, this combination allows database access as well as the possibility to develop interfaces and implement complex algorithms. The main features are:

– **Product definition**: the creation of a new product to allow retrieval and modifications of its information.
– **Output variables**: Designers are asked to define functional areas on the PCB. These areas will be used as outputs by the linguistic model.
– **Loading components**: The positions of individual components on the PCB are retrieved automatically using CAD files.

- **Components definition**: Designers are asked to specify association between components and functional areas.
- **Input variables**. Designers have to describe the different measurements obtained from functional testing. They also have to assign these measurements to their associated functional area.
- **Limits definition**: Control limits for each test are used for calculating the membership definitions.
- **Operator interface settings**: The operator interface uses a picture of each PCB extracted from CAD files. However, pictures are not standard and both triangulation method and scaling are used to obtain an overlay between the picture and component position.
- **Calculation of the polynomials and the equations**: This one-click button is used to compile all previous information and generate the polynomials as well as the linguistic rule base.

The main criterion for this software is to be simple to use for operators. This goal has been achieved by reducing the amount of information requested from operators to its minimum. When the operator selects a product number from a list and the test number of a faulty PCB, the system provides defect feedback for that PCB as shown in Fig. 17. Feedback about possible defects is given to the operator in a very simple and visual way that allows fast comparison between the real PCB and the picture on the screen. A red cross marks possible defective components and provides defect description.

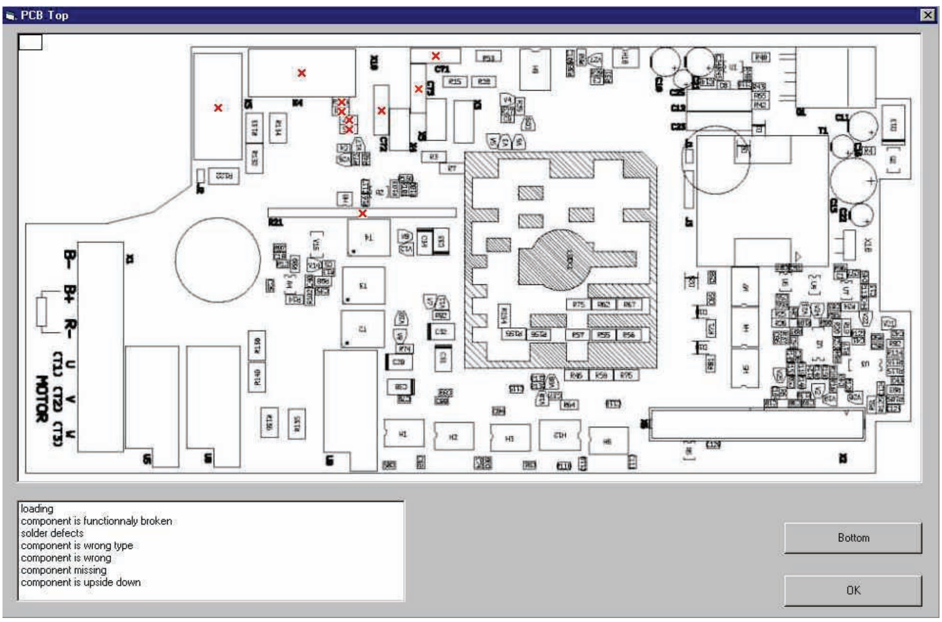


Fig. 17. Operator software (Gebus & Juuso, 2002b).

4.4.5 Results

The first tests of this decision support tool were made on a widely used product. This product had been produced already seven months at the time of the tests and measurements were available for around 280 products. This means that the polynomial regression and equation calculations were made with 280 measurements for each variable. Out of those 280 products there were 42 PCBs with errors. When tested with those 42 PCBs, the system gave the right answer for 95% of the cases. When adding faultless products into those tests, a rate of good answers slightly over 95% was obtained.

Next, products in a prototype stage, with very few data available, were tested. With data from 15 products, representing much less than one day of production, the rate of good answers was 90%. It was therefore possible to assume that the ramp-up period for the system represents less than one day's production.

Robustness has been another factor checked during tests. Besides working well even with new products, the results with products known to have problems as well as others with very high yields were tested. By adjusting the corrective factor, it was always possible to get good results. The same comment could be made about the effect of different kind of measurements and both analog and binary measurements could be used successfully.

The primary objective of helping operators to perform repair actions on faulty products has been reached by providing defect localization in a simple and efficient form. Fig. 18 shows how this system interacts with its surrounding environment. Defect localization has been achieved using an LE-based approach combined with functional testing as well as expert knowledge. Although defect localization concerns only functional areas of a PCB, the system provides sufficient help to operators who do not need anymore to "learn" every new product.

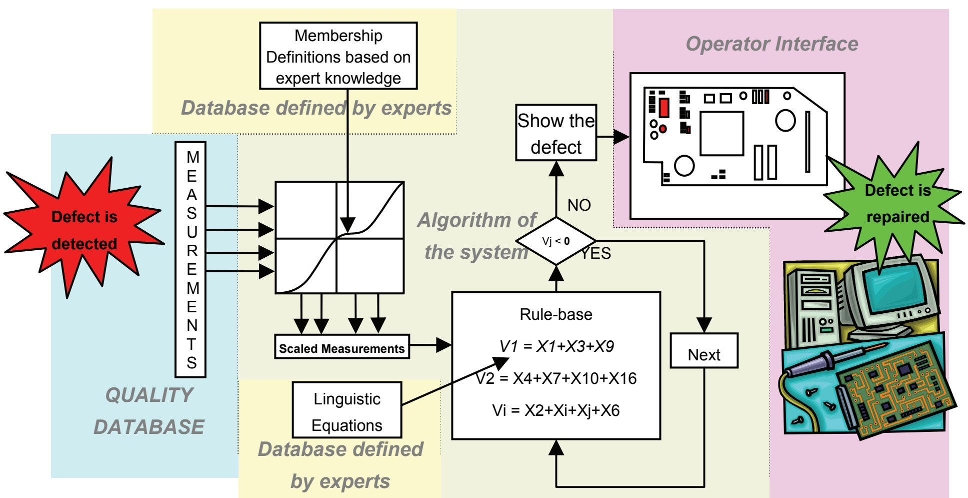


Fig. 18. General structure of the Decision Support System (Gebus & Juuso, 2002b).

Component level localization is currently not possible with LE because of an insufficient amount of defect data. However, the self-tuning ability of the system makes component level localization a potential improvement in the future. In the meantime, another suggestion for improving feedback is to use a simple probabilistic approach to the localization problem by applying a case-based reasoning to previous similar defect cases.

4.4.6 Advantages and disadvantages of the LE approach

Firstly and probably the major interest in the method is how easy it is to implement. Membership definitions are calculated with two polynomial regressions with three points out of five fixed. Therefore, linguistic equations can be defined in advance. Once the resulting linguistic rule-base is available, implementing the system consists of two steps. The first step is to fuzzify real values by applying functions of the type presented in section 4.3.1. The second step is to run the linguistic inference engine on the resulting linguistic variables, which basically consists of checking results of inequalities. This process is fast and can be used on-line as a real-time system.

As seen from the results in this case study, the method becomes sufficiently reliable after a very short ramp-up period as well as showing robustness to various cases. This can be explained by the fact that the system is configured with almost only parameters defined by “experts”. Tuning becomes therefore a very simple task since only few parameters can be modified. The addition of expertise is in effect simplifying the LE system allowing self-tuning ability with the click of a button.

Unfortunately, this last point is probably also the biggest limitation to the system. By not relying enough on experimental data, only approximate knowledge is available. Although it is not primordial in fault diagnosis to find the root causes of a defect or even to separate defects from each other, a consequence of this is that it is also impossible to distinguish defective components from each other. However, even if the localization of a defect on the PCB is limited to an area composed of 10 to 20 components, this represents a major improvement considering that there can be hundreds of components on a PCB.

4.5 Conclusion

Fault diagnosis usually aims at identifying root causes of problems so that corrective actions can be taken. The electronics industry, however, represents a specific context in which the notion of zero defects is, at least from a statistical point of view, a utopia. Furthermore, because the price of an individual component is negligible, the real aim of fault diagnosis in the electronics industry is limited to the localization of a faulty component so that it can be replaced.

Different approaches to the fault localization problem are possible. However, without extensive knowledge being available, many of the proposed knowledge-based techniques fail to achieve sufficient performance. Automatic generation of systems, model-based techniques and adaptation techniques are very valuable as a part of a hybrid approach for developing and tuning systems for fault diagnosis.

In particular, the LE approach originating from fuzzy logic is very efficient in solving these problems through its modeling technique: models can be generated from data, various types of fuzzy rule-based models can be represented by LE models, and any LE model can be transformed to fuzzy rule-based models. Linguistic equations are also useful for taking into account non-linearity especially in multivariable applications.

Furthermore, addition of expertise, far from complicating the approach, is in effect simplifying its implementation, whereas the inherent self-tuning ability allows efficiency improvement of the system during its use. This is illustrated in the case study where linguistic equations are combined with expert knowledge to successfully detect and trace a defect into a small area of a PCB. If a sufficient amount of data is provided, self-tuning and self-learning methods could be used to improve the efficiency of diagnosis to component level localization.

Finally, the LE approach represents a methods integrating technique which makes it a good candidate in the larger decision support system framework, along with knowledge management techniques and defect forecasting methods.

5 Fault detection and short-term statistical feedback control of an electronics process

Quality is a concept that has been around for a very long time and lack of it could have had dreadful consequences for some people. Egyptians used to kill architects if the houses they designed collapsed. Although the previous chapter has shown that the concept of zero defects is a utopia, especially in electronics manufacturing, this example shows that already during ancient times the need to ensure reliable processes existed and not just rely on final check by the customer.

Nowadays the consequences of non-quality can be in many ways even more severe. Loss of customers and image degradation of a brand affect equally all employees in a company, not just the one who did a lousy job. In fact, the entire supply chain will be affected through the resulting snowball effect. Therefore, quality needs to be checked at every step of the design, development, and manufacture of a product.

General approaches as well as context-specific methods are necessary to prevent any process from shifting or becoming uncontrollable. Continuous improvement aims at reaching this target and requires two types of activities that will be the topic of this chapter. First, process conditions have to be monitored so that the behavior of the system can be forecasted. Secondly, decisions need to be made based upon the estimated behavior so that uncontrolled situations might as much as possible be avoided.

5.1 Continuous improvement

Continuous improvement is a phrase suggesting that a process or product should always get better as knowledge about it and experience with it accumulates over time. The following part of this chapter is an introduction to some concepts and tools used in continuous improvement. Additional information can be found in any good textbook related to quality such as in Oakland & Followell (1990), from encyclopedia such as Wikipedia (http://en.wikipedia.org/wiki/Quality_Assurance), or from organizations such as the American Society for Quality (http://www.asq.org).

Quality assurance is at the heart of the continuous improvement approach. Although they have different meanings, the terms "quality assurance" and "quality control" are often used interchangeably to refer to ways of ensuring the quality of a service or product. Quality assurance represents the planned and systematic activities aimed at giving confidence that quality requirements for a product or service will be fulfilled, whereas quality control aims at observing and guiding a process to fulfill requirements for quality (http://www.asq.org/learn-about-quality). Quality assurance covers therefore all activities from design, development, production, installation, servicing and documentation. It is about setting up systems to make the work more repeatable, so that even if mistakes happen, at least the same mistakes are made each time. The point is that repeatable mistakes can be corrected and eventually, quality assurance will lead to a system in which mistakes have been eliminated. The ISO 9000 series of standards, for example, define models for quality assurance including business practices in different industries.

Under the quality assurance approach, suppliers are responsible of upstream quality and must ensure it before the shipment to the customer. A supplier audit is used as a way to certify their capability to ensure their own internal quality. But certification is not enough to achieve the supplier's ability to deliver cheap and reliable items. Efficient communication and collaboration at every level of the client/supplier relationship are also necessary, especially in industries like electronics where there is a high use of subcontractors.

From the author's experience over the past few years, however, this necessary mutual trust is rather an exception among the encountered players. Not only are these relationships not running deep enough, but the situation is made worse by subcontractors sometimes hiding their ignorance about quality control behind confidentiality agreements. These agreements can become counter-productive, if their purpose is to elude any responsibility. This attitude is unforgivable considering that many quality assurance methods exist and have been adapted even by non-manufacturing organizations as a way of sharing expertise, resources, and risk in a complex global environment. The most widely-used tools for continuous improvements and quality assurance are:

- **Plan-Do-Check-Act (PDCA) cycle:** Also known as the Shewhart cycle, it is one of the most widely used methods to coordinate continuous improvement efforts. A PDCA cycle consists of identifying the problems and defining changes or tests aimed at improvement, carrying out the change or test, studying the results, and adopting or rejecting the change.
- **Six Sigma:** It is a fact-based, data-driven philosophy of quality improvement that values defect prevention over defect detection through reduction of variation and waste. In practice, achieving Six Sigma quality performance means that there are no more than three or four defects per million opportunities.
- **Total Quality Management (TQM):** TQM is an approach first introduced in 1985 combining Statistical Process Control (SPC) principles and quality management methods to maximize potential benefits when undertaking performance improvements. It is considered a management strategy to embed awareness of quality in all organizational processes and aiming at long-term success through customer satisfaction.

5.2 Statistical process control

Industrial processes are subject to many known or unknown malfunctions during their operational life times. Obviously, malfunction will reduce the efficiency, deteriorate product quality, and damage the equipment. Many organizations use statistical control to bring the production to the six-sigma levels of quality. This control is usually done by randomly sampling and testing a fraction of the output. Whatever statistical method used, they all originate from the 1920s and the work of Dr. Walter Shewhart, a statistician working for Bell Laboratories who was asked to find out why there was so much variability between the telephones they produced. Focusing first on the quality of produced goods, his approach moved from statistical quality control (SQC) to statistical process control (SPC) when integrating a more proactive attitude toward the production tools. SPC, described in details in Oakland & Followell (1990) has been used rather successfully over the last decades to improve the output of manufacturing processes.

SQC determines sampling plans and acceptance levels for a given lot based on consumer and producer risks, lot sizes and acceptable reject levels. Although it guarantees that the quality of delivered goods will conform to a predictable level, it cannot signal where or when a problem has appeared because it is always applied at the end of the process, when the problems have already occurred. Therefore, SQC alone is not an approach that aims at improving the process. It is only an informative tool about the overall quality of a lot.

SPC, on the opposite, uses in-process inspection to discover potential problems. Sampling is done frequently, which increases the chance of finding problems in the early stages of a process, thus limiting waste and disruptive effects on the production. The aim is to monitor the process itself to reduce (but not remove) the need for inspection of every output. The benefits of SPC over SQC are obvious:

- SPC retains at least SQC requirements.
- Faster feedback because problems are discovered as soon as they occur.
- Production people are more involved in quality since they are also the ones verifying it and taking corrective actions. Thus separate inspectors can be removed.
- Information from the process is obtained with more details.

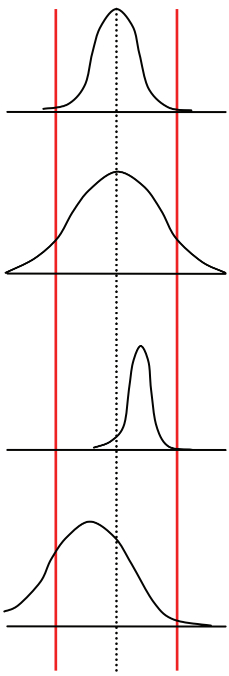
More recently SPC went even one step further by integrating engineering process control tools, which regularly change process inputs to improve performance. This Run-by-Run approach will be described in section 5.3.

SPC originates from the “parts” industry which typically attempts to reproduce individual items as accurately as possible. The aim is to prevent the manufacture of defective products by controlling the input, settings and parameters of a process as tightly as possible so that the outputs will be under control. Dr. Shewhart's research led him to the conclusion that every process displays variation that can be partitioned into two components:

- Natural process variation is the variation inherent to the process. It is a controlled but unavoidable variation that shows a stable and consistent pattern over time. This variation is due to common causes affecting the process.

– Special cause variation is typically caused by some problem or extraordinary occurrence in the system. It is uncontrolled and shows a pattern changing over time. This form of variation indicates that the process itself is unstable, hence out of statistical control.

Through observations and measurements, SPC tries to separate random variations due to common causes from non-random variations that will have to be corrected. To do so, control limits are set to determine if a process is "in control" or “out-of-control”. This brings about the question of process capability, or whether the process is able or not to generate an output within desired control limits. Fig. 19 illustrates this problem by giving a representation of the four most common situations that can be encountered when running the statistical analysis of the process output. In these various situations, first thing to do is usually to bring the process back in control. If the process is nevertheless incapable of producing an adequate level of quality, attempts must be made to identify and reduce the common causes of variation.



Process “in control” and capable

Process “in control” but not capable

Process capable but “out-of-control”

Process “out-of-control” and not capable

Fig. 19. Output distributions for different kind of processes (Gebus, 2000).

5.2.1 SPC tools

Control charts, originally developed by Walter Shewhart in the early 1920s, are without contest the most widely used SPC tool. They are used to study how a process changes over time and present data in a way that clearly indicates when action is necessary, when further information is required, and when no action is needed. In other words, control charts attempt to distinguish between common causes and special causes of variation. The idea of the control chart is 'proactive', in the sense that it is intended to monitor processes when they go 'out-of-control' and thereby ensure quality products.

Many kinds of control charts exist and the choice will depend on the process characteristics and the kind of variation that should be detected. All control charts, however, have some common features, like a central line for the average, and upper and lower lines for the control limits as shown in Fig. 20. These lines are determined from historical data. In practice, control limits are often drawn at ±3σ and sometimes, warning limits are added at ±2σ. Control schemes, however, can take many other forms.

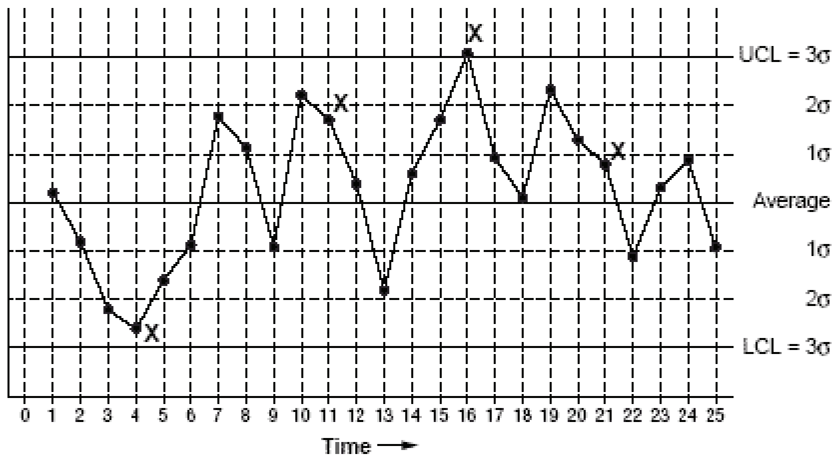


Fig. 20. Example of a control chart.

Stuart *et al.* (1996) made a good review of the different control charts available. Mean and Range charts (Shewhart charts) are most suitable in the context of mass production where data is available on a large scale. Although they are less efficient than other charts for the purpose of detecting changes, they are easy to implement and understand since they reflect in a rather straight way the variation of the process.

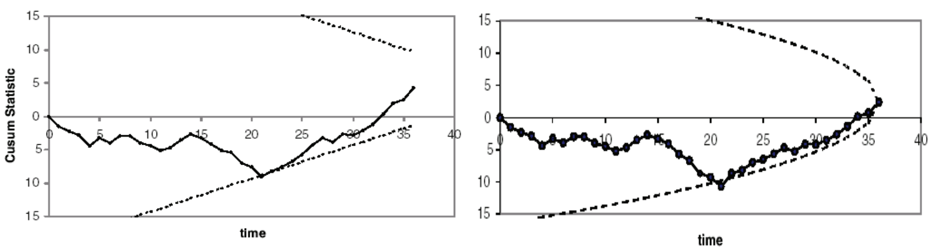
A mean chart makes use of the central limit theorem to state that a sample mean should be normally distributed around the steady state mean of the stable process. It is therefore easy to calculate 2σ and 3σ limits, and use them as warning and control limits. Furthermore, the sensitivity can be tuned by adjusting the subgroup size.

The range chart monitors the variation of the process. Since the control limits on the mean chart depend on this variation, it makes no sense to analyze the mean chart if the range chart is out-of-control.

When it is not possible to use mean charts, then individual charts are usually employed. Under such circumstances, median charts can offer an interesting alternative. Like mean charts, median charts have the advantage over individual charts that they can be made more sensitive by increasing the subgroup size. Thus, median charts can be used to fill the space between individual charts and mean charts. They do have the disadvantage of not taking into account the extent of the extreme values that can provide useful information for control purpose. The method used in Case Study No. 3 proposes a solution to this problem by monitoring extreme values in addition to the central value.

One of the problems when using individual charts is to take a decision mostly based upon the last measurement without taking into account the larger evolution of the process. Moving average charts have been developed to solve this problem by grouping data together over short moving time windows. The exponentially weighted moving average (EWMA) chart, originally proposed by Roberts (1959), is an extension to moving average charts. It is based on the idea that old data might be outdated and should weigh less on the control decision than more recent data. Crowder (1989) distinguishes between the two different uses of the EWMA. First, it allows forecasting the values of a process mean. Secondly, it has the ability to detect very small shifts in the average values.

Originally proposed by Page (1954), the cumulative sum (CUSUM) chart is a tool that can be used to detect small but steady shifts of a process output. Each value represents the sum of the previous value and the most recent sample mean deviation from a target value. If the process keeps in control, the cumulative sum describes a random variation. On the other hand, if a monitored average changes to any value above or under the expected average, then an ascendant or descendant tendency will appear. To determine if a process is shifting, the usual method is to apply a mask on the CUSUM chart as in Fig. 21. If any previous points fall outside of the arms of the mask, a signal is generated.



V mask   Parabolic mask

Fig. 21. Different masks for detecting abnormalities with CUSUM charts (Runger & Testik, 2004).

CUSUM charts are more efficient than Shewhart charts in detecting small process changes. Furthermore, if a problem is known, it is possible to create a test for that particular cause. However, this chart might be useless to detect unexpected forms of disturbances. Goel & Wu (1973), for example, present a procedure to determine the optimum values of the samples size, the sampling interval, and the decision limit for CUSUM charts to control the process average when the values are normally distributed. Morgenstern *et al.* (1988) propose a modification to the standard CUSUM approach in which the instant of failure and its duration are established. With this method, anomaly detection is improved and the anomaly can be characterized.

For Hunter (1986), the differences among Shewhart, CUSUM and EWMA control charts have to do with the way each charting technique uses the data generated by the production process. A Shewhart chart depends entirely on the last demarcated point. A CUSUM chart attributes equal weight to the most ancient datum as well as the most recent. The EWMA gives a higher weight for more updated information and lower weight for more remote information. The non-necessity of normally distributed data weighs in favor of the EWMA over the CUSUM, though as with all control charts, the assumption of independent subgroups needs to be investigated.

Most of the control charts described previously do not use individual data. SPC actually recommends to subgroup the data so that the central limit theorem applies to the subgroup means and hence the means will be approximately normally distributed. Sensitivity to non-normality raises practical problems such as false alarms. On the other hand, individual control charts should not be systematically discarded because they carry more information about the process, whereas a mean chart, for example, hides what the process actually looks like. This loss of information can result in lower process control efficiency. When it comes to the decision-making process, it is always better to have a picture as accurate and complete as possible, and the only way to do this is to look at the actual distribution of the individual observations. The choice between individuals and mean charts should therefore be made according to the cost of inspection relative to the cost of failure.

In any case, knowledge about the process should be used for creating charts that better express process variation. If enough knowledge is available, it can be used to enhance mean or median charts as for example in Case Study No. 3 in which the control information can be retrieved from grouped data.

5.2.2 Control schemes

SPC is a methodology for charting the process and quickly determining if a process is "out-of-control" by separating common causes from special causes of variation. However, deciding when a process is really out of statistical control is not necessarily as straightforward as it seems. Indeed, an output value can be beyond control limits consequently to common cause variation, as a process can be out-of-control even if no value has yet shifted outside the control limits. Various tests have been developed to help determine when an out-of-control event has occurred. However, as more tests are employed, the probability of a false alarm also increases.

When a process is out-of-control, the first special causes of variation have to be found so that the process comes back into statistical control. Secondly, through process improvement, the inherent variation can be reduced. This typical cycle is illustrated in Fig. 22 by the narrowing of the control limits toward the centerline of the process.

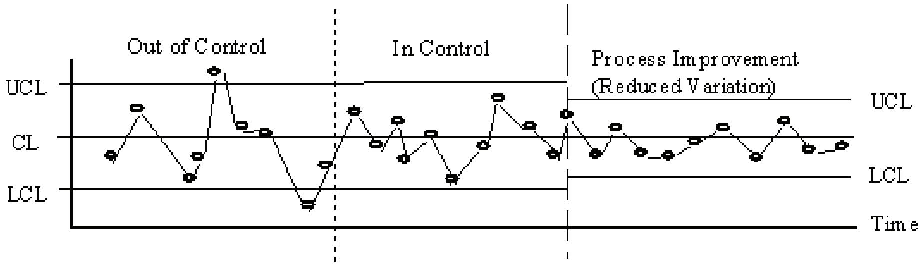


Fig. 22. General steps for process improvement with SPC approach.

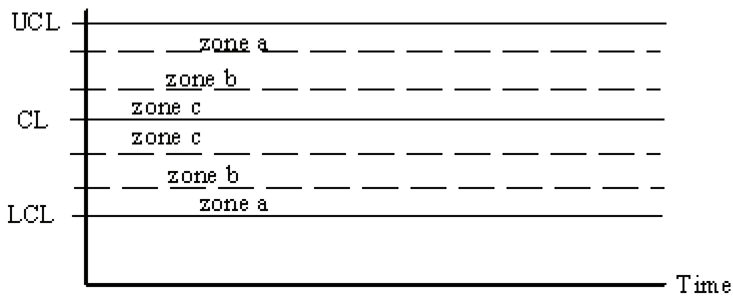


Fig. 23. Control chart zones.

Several types of conditions exist that indicate that a process is out-of-control. Before going into more complex problems, basic tests are usually sufficient. A control chart can be broken down into three zones, a, b, and c as in Fig. 23. A series of rules can then be defined to detect conditions in which the process is behaving abnormally, thus suggesting that it is out-of-control (Oakland & Followell, 1990). Although most of these rules are based on a very simple analysis of variables, these have provided efficient means for controlling industrial processes during the past decades. Furthermore, as will be shown in Case Study No.3, simple techniques can still provide successful solutions even in today's more complex environments.

5.2.3 Considerations for a modern SPC

In recent years, new challenges have arisen in electronics, driven not only by changes in products, but also by changes in working habits and mentalities. The increased complexity of manufactured goods has lead almost mechanically to an increase of the number of sources for variation, thus to an increased pressure on quality requirements for individual parts. Modern technologies can partly counteract this effect through enhanced machine capabilities enabling, for example, to switch from 3σ control limits to 6σ control limits. As a result, out-of-control situations or individual defects are not anymore counted in parts per millions, but rather in parts per billions. Technological advances, however, can only have a limited impact on quality, especially if they are taken individually. The future lies therefore in more global approaches to quality in a much larger sense.

Modern quality control needs to provide solutions that are reactive, robust, and flexible because companies are not able to engage in process adaptation, even when their benefits are clearly estimated. Therefore, modern SPC approaches need to take into account that manufacturing processes are what they are. They must make the best use of the fixed framework with few parameters, sometimes incompletely monitored, and most often not relevant to manufacturing problems. Based upon this idea, a series of challenges that are described in the following part of this section have been defined.

Short-run and short-term problems

A major drawback of SPC-based inspection strategies is their dependency on the stability of the process. They are traditionally applied to processes that are continuous by nature, which means that production runs of the same product are in quantities of hundreds or thousands. However, modern manufacturing tends to move more and more toward flexibility and just-in-time production. This brings out a problem when attempting to use traditional Shewhart charts because there is never enough data to calculate meaningful control limits. As a result, they are not well adapted for real-time or even short-term feedback control.

For that purpose, Cullen & Bothe (1989) developed a new generation of short-run control charts that allow the operator to plot different part numbers on the same chart and detect more easily time-related changes in the process. Nugent (1990) also attempts to implement SPC in short-run processes by using individuals-moving range charts, target charts, or target individuals-moving range charts instead of mean and range charts.

In the more specific framework of discrete event systems, Paoli & Lafortune (2003) explain that the safe diagnosability of these systems needs that every failure event of the process leads to observations distinct enough to enable unique identification of the failure type with a finite delay. They point out the problem of performing the detection before the system executes a forbidden string.

False alarms

Both fault detection and diagnosis are extremely important to modern industrial processes and many methods exist. Process engineers might encounter problems when using SPC to monitor non-normally distributed variables because of the high rate of false alarms and the inability to detect trends with classical charts. Cuéllar (1991) proposes to modify the distribution of a non-normal variable so that it becomes normal. Then all the well-known and accepted methods of SPC can be used to monitor these variables. One drawback is that the ordinate of the SPC chart is in the units of the transformed variable.

Even when the variables are normally distributed, the measurement process of the control parameter still involves random disturbances that arise from the environment or the measuring instrument itself. Because of these disturbances, the control parameters may exceed the predetermined control limits and generate false alarms. This problem is addressed by El-Shal & Morris (1999, 2000) through the use of an additional fuzzy rule-based method.

Without having to come to hybrid methods, Margavio *et al.* (2004) compare different methods for calculating control limits for EWMA and CUSUM charts on the basis of their false alarm rates. They argue that control charts can be more flexible when designed according to desired false alarm rates and average run lengths.

Incomplete data

Some machine learning techniques have the ability to learn from data and to handle uncertain and imprecise situations. Similar approaches can be applied to data aimed at statistical analysis. Kang & Park (2000) suggest integrated methods of inductive learning and neural networks to solve quality problems.

Some control problems are characterized by the presence of uncontrollable as well as unobservable events, which represent another case of incomplete data. Marchand *et al.* (2001) propose a solution based on the notion of occurrence, control costs, and a worst-case objective function.

Disordered data

Collected data is not always relevant for the problem-solving purpose, even when the right data is collected. For example, the sequence of end-of-line data is not always the same as the sequence in each process step, thus an abnormal trend in any of the process steps is more difficult to detect based on the end-of-line data than based on single process data. In Case Study No. 3, a solution is proposed that can handle data with long-term batch disorder by taking into account only short-term data for the analysis.

In the case of a higher degree of disorder, more complex methods are required as eliminating the sequence-disorder or multiple-stream effects through brute force might not be feasible. Fan *et al.* (1997, 2000) show that by applying EWMA to sequence-disordered data, a moving average can smooth out the sequence-disordered effect and weighting factors, allowing the choice of an effective window size so that the underlying trend can be popped out. With this approach, fewer control charts are needed for each parameter as compared to those needed for the "brute force" SPC application. Furthermore, in-depth diagnosis is only performed at the critical process steps. As a result, the root causes can be discovered much more efficiently.

Chen *et al.* (2000) propose a method to arrange and group sample data to properly construct and interpret the control charts by taking into account different sources of variation through innovative data sampling. First, an analysis of variance allows decomposition of data into significant sources of variation that should be watched more closely. Secondly, these sources, once rearranged and regrouped, are analyzed by four multivariate control charts.

Multiple variables or correlated variables

Classical SPC considers individual and independent quality measurement sources whereas most modern electronics processes are multi-input and multi-output. Consequently, it is difficult for an operator to implement effective control, and SPC results only in binary control actions as described by Ha *et al.* (1990).

Multivariate statistical process control methods, such as the multivariate control-hotelling's T2 Chart, have been developed to address some of the limitations of univariate monitoring techniques by considering all the data simultaneously and extracting information on the 'directionality' of the process variations.

The most practical approaches to multivariate SPC appear to be those based on multivariate statistical projection methods such as PCA and PLS as described by Martin & Morris (1995), and Martin *et al.* (1996). The idea is to reduce the dimensionality of the problem by projecting the multivariate data down onto a lower dimensional space, which contains all the relevant information. Wilson & Irwin (1998), followed by West *et al.* (1999), propose a nonlinear extension of the multivariate approach using a series of radial basis function neural networks. As a result, the principal direction will be replaced by a principal curve which minimizes the orthogonal deviations from all of the original variables.

On a slightly different problem Champagne & Monette (2002) describe how multivariate batch SPC models can be built using two levels, an observation level and a batch level. The lower level describes the dynamic time dependencies of the batch process, while the upper level focuses on how each completed batch varies over time.

5.3 Real-time process control

It is management's responsibility to reduce common causes as well as special causes of system variation. This can be done through process improvement techniques, new technologies, or reengineering the process to have fewer steps. If a process shows low variation, it will be more predictable which makes it easier to keep outputs close to a desired value.

Traditional SPC measures the performance of the process over time and applies statistical techniques to diagnose control. However, these statistics are based on the assumption of independent samples coming from a common probability distribution. As a result, variables must be held constant otherwise their probability distribution may change over time and no statistical conclusions can be drawn. This constitutes a serious restriction on the operation of the process since it forbids any form of process tuning during its operation. Therefore, traditional SPC often leads to a process improvement breakthrough at the expenses of optimal operations. Processes are run with fixed settings over several hundreds of batches, and only occasionally re-tuned.

The electronics industry has an ever-increasing need for precision in their manufacturing processes. An alternative approach is to use feedback control techniques that use measurements during processing to adjust process settings in real-time.

5.3.1 Feedback control

Most electronics manufacturing equipment is still designed to work in an open-loop mode. Due to this, the manufacturing performance is not as good as it could be. The ability to produce highly customized products necessitates new kinds of flexible manufacturing systems. The complexity of flexible manufacturing processes makes the supervision and maintenance task difficult to perform by human operators as described by Seabra Lopez & Camarinha-Matos (1995). The authors describe a planning strategy and domain knowledge for error recovery. Supervision architecture provides functions for dispatching actions, monitoring their execution, and diagnosing and recovering from failures. Through the use of machine learning techniques, the supervision architecture is given capabilities for improving its performance over time.

Statistical techniques, when applied to process control, have often been used for dealing with quantitative observations. Qualitative observations that represent subjective knowledge about the process (*e.g.* an operator's opinion about a problem) have been ignored. This is especially unfortunate since qualitative observations encompass usually the means to adjust a system. Spanos & Chen (1997) propose to integrate qualitative information for process tuning purpose arguing that they better explain relationships between input process settings and the process output response. The authors propose a framework for modeling such qualitative process characteristics by developing an intelligent computer-aided manufacturing system that can capture qualitative as well as quantitative aspects of a manufacturing process.

In terms of process adjustments, Pan & Del Castillo (2001) compare process adjustment methods and their integration with SPC charts to provide relevant feedback to the production tool.

5.3.2 Run-by-Run approach

Run-to-Run (R2R) or Run-By-Run (RbR) control has been developed at the MIT as an answer to the basic stability assumption requested by traditional SPC approach. It enables tweaking of equipment settings between runs in order to optimize the operation of equipment, and to minimize process shifts and drifts. RbR control can therefore be seen as a process control technique for batch-processing environments in which an algorithm is used to adjust process parameters prior to each batch. Its effectiveness has been demonstrated in a variety of processes. Moyne *et al.* (2000) explain how information from previous runs, prior process inputs, or environmental information might all play a part in the control algorithm. They present 20 contributions that provide a practical guide to the understanding, implementation, and use of RbR control in electronics manufacturing and manufacture in general.

RbR control works by continually updating a model of the process and designing a new recipe based on that updated model. Working in harmony with the controller is generalized SPC, which permits the diagnosis of a process as it is being tuned. Together, RbR control and generalized SPC provide an approach to automated on-line optimization and control of processes as described by Sachs *et al.* (1990). RbR implements a form of adaptive control based on the sequential design of an experiment that is similar to the way a process engineer operates a piece of equipment. The controller automates and improves the traditional actions of the process engineer by creating a mathematical formalism that allows multi-parameter responses to multiple attributes. Sachs *et al.* (1995) describe how SPC and feedback control can be combined. A good RbR controller should be able to compensate for various disturbances, such as process shifts, step disturbances and model errors. Moreover, it should be able to deal with limitations, bounds, cost requirement, multiple targets and time delays that are often encountered in real processes as explained by Zhang *et al.* (2000). Generally there are two modes in which an RbR controller can operate:

1. **Gradual mode:** This mode responds to gradual drifts in the process such as those caused by tool wearing. The purpose of the gradual mode is to compensate for drift in process by gradually updating a model for the process and prescribing a corrective action based on that model.
2. **Rapid mode:** Some events can cause sudden shifts in the process and the rapid mode of the RbR controller deals with such situations through rapid modification of the process model. These situations include maintenance operations, switching among different specifications, and detected shift in standard processing condition.

Fig. 24 shows the general structure of the RbR approach. The choice between the two modes is determined by the outcome of generalized SPC to be applied to a process while it is being tuned as well as by the circumstance. The deadbeat controller described by Boyd & Banan (1995) represents the ideal case when corrections can be done completely from one run to the next. In an experiment at the MIT, rapid mode recovered the process within 3 runs after a disturbance whereas gradual mode reduced the variation of the process by a factor of 2.7 as compared to historical data.

Table 1 shows the main differences between the traditional SPC and the RbR approach. Performance benchmarks given by Hu *et al.* (1992) show that the RbR controller combines the advantages of both SPC and feedback control to provide an accuracy and flexibility of control that cannot be obtained by using other methods. It can also be applied to an N-Dimension linear system and the use of post-process measurements for control purposes can compensate for the lack of appropriate sensors for real-time control.

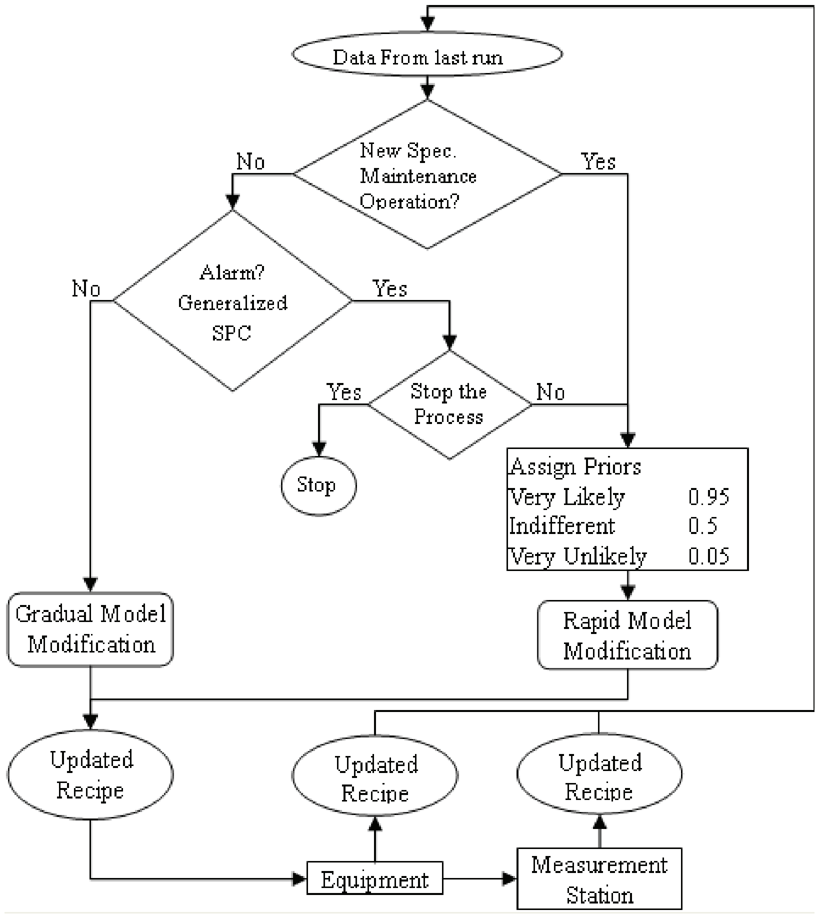


Fig. 24. Flowchart of a RbR approach (Hu *et al.*, 1992).

Table 1. Comparison between SPC and RbR (Sachs et al., 1990).

SPC	RbR Controller
No Tweaking	Continuous tweaking
Controls, but does not optimize	Optimizes and controls
Detects shifts, but response is up to operator	Compensates for shifts and drifts
Does not accommodate slow drifts	Multivariate responses to multiple outputs

It is shown that sequential adjustments are superior to single adjustments for almost all types of process shifts and magnitudes considered. Pan & Del Castillo (2001) argue that a CUSUM chart used together with a simple sequential adjustment scheme can reduce the average squared deviations of a shifted process more than any other combined scheme studied when the shift size is not very large.

Another work presented by Boning *et al.* (1995) is based on the RbR control of chemical mechanical polishing. The issues include limits on multiple input variables as a fast heuristic approach constraint method and a full optimization approach are compared. An input weight method enables the process engineer to choose which input parameters should be more readily modified and which ones should be changed less. An EWMA controller determines the inputs for the next run of the process. In this method, control actions are only taken when the EWMA controller detects a significant mismatch between the dynamic model and the actual process.

Del Castillo & Rajagopal (2002) propose as well to use an EWMA to enhance a "predictor-corrector" feedback controller. The controller is based on two coupled multivariate EWMA equations. The proposed double EWMA feedback controller is compared to the common practice of using multiple single-input-single-output double EWMA controllers running in parallel and it is shown that the stability conditions are invariant with respect to various realistic drift disturbance models.

In addition, on the topic of multivariate processes Chaudhry *et al.* (1996) propose to apply fuzzy sets theory to develop a multi-algorithm control system that can exploit the information from multiple sources to make better control decisions. Generic decisions are developed to make techniques capable of utilizing relatively contradictory information from heterogeneous sources. The developed controller provides better RbR control by determining what combination of the available control algorithms should be invoked to obtain the optimal response for the control of the process.

5.4 Case study No. 3: a short-term process control tool

Today, the production technology has evolved to the point where new and complex SPC methods are required. However, simple approaches should not be neglected. This case study, which was the object of a publication by Gebus & Ruusunen (2004), proposes therefore a simple approach to some of the problems presented earlier. The first step characterizes the environment in which the approach has been developed as well as the constraints and available tools for the problem. This will allow the presentation of the concept of a supervision system based on the calculation of few quality indices enabling efficient process monitoring. Details of calculation with clear examples and verification on the limits will be given. Finally, implementation issues will be discussed with an emphasis on feedback provided to line workers.

5.4.1 Context

In this case study, the production line is producing prototype items and quality control is performed using automated optical inspection (AOI) between placement machines and the soldering oven. This means that all the components are placed on the printed circuit board (PCB) when the board is being inspected. The AOI system measures the position in X and Y directions for each component. The control system provides information on missing or misplaced components according to memorized patterns that have been learned from a test PCB. However, this does not occur without generating false alarms due to imprecision in the measurements or imperfections in the test PCB.

Because of the high level of false alarms, all PCBs are double-checked by a human operator. Furthermore, since no feedback information is available to them, operators may have to walk up to several hundreds of meters as the information required to take corrective actions is scattered over several pieces of equipment and can even require the operator to leave the factory floor. Considering the fact that the production consists mainly of prototypes or other small batches, corrective actions often come too late as the chain of actions needed for taking corrective actions is too long, thus inefficient. In the flowchart shown in Fig. 25, only a small number of actions can be considered as productive.

The production line studied is producing from small to very small amounts of similar products. The production diversity is high and a certain type of PCB may be produced only on weekly or even monthly bases during 15 to 30 minutes. This implies problems such as disordered or short-term data described by Chen *et al.* (2000) and Nugent (1990). On one hand, any control tool will have to deal with short-term analysis and feedback in order to provide quick response to the production line. On the other hand, analysis based on history data will have to take into account the fact that this data comes from very different batches. From one batch to the next, a large amount of parameters can increase data variability. For this reason the dependency on history data that can be outdated has been minimized.

In this case study, data has been compressed before being stored onto a database. This usually destroys the essential nature of the process data and eliminates much of the useful information as explained by MacGregor & Kourti (1995). Only incomplete data is available for analysis. On a slightly different topic, Marchand *et al.* (2001) also discuss the case where some variables are unobservable or uncontrollable. The aim of data analysis is to extract again useful information from the compressed data.

Measurements taken from the line are concerned with the component misalignment. In the case when several occurrences (sometimes up to 100) of the same component can be found on a PCB, only the average, minimum and maximum misalignment values for the last 20 PCBs are saved to the database, reducing drastically the size of it. In addition to this, modifications have been made to save also the reference number of components considered to be responsible for the minimum or maximum misalignment. This is the only data available to a supervision system for analysis and control.

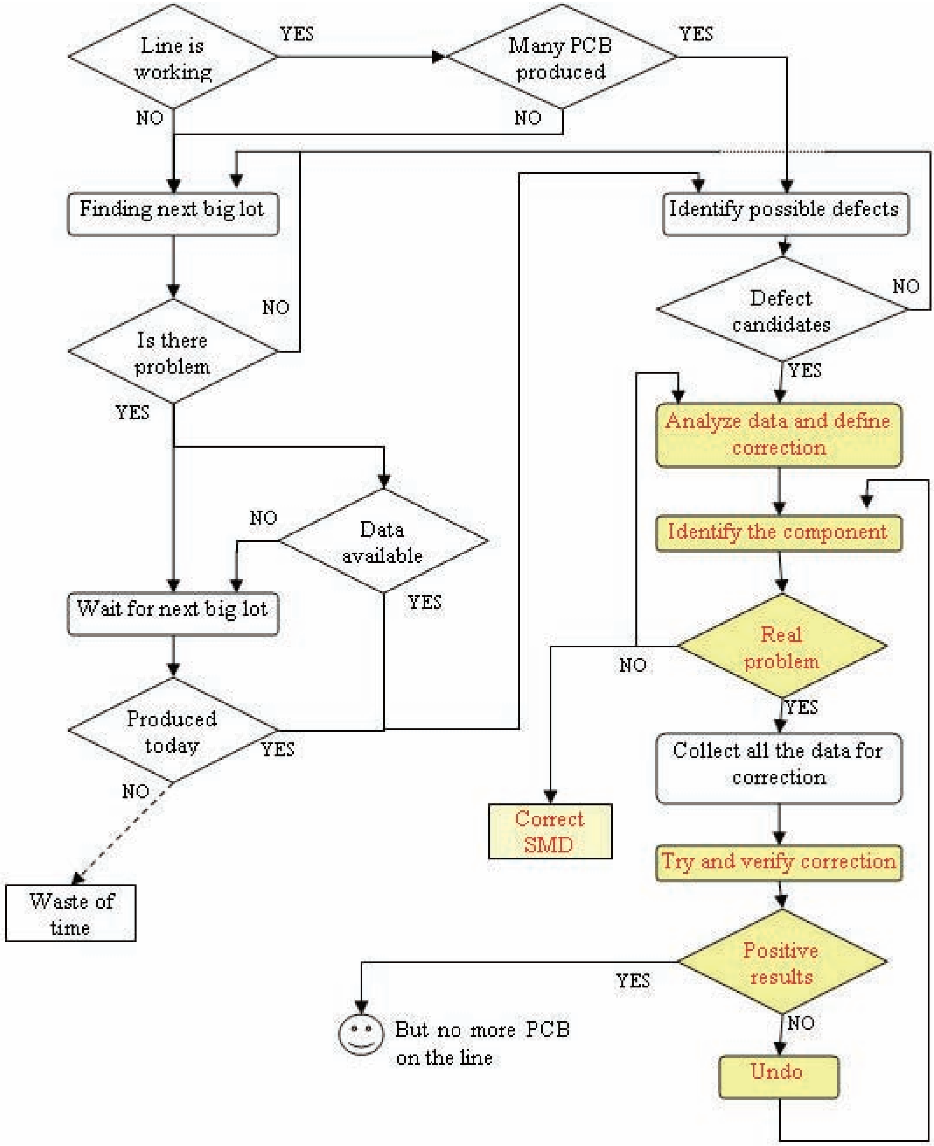


Fig. 25. Flowchart representing the procedure for implementing a corrective action in the beginning of the case study (Gebus, 2000).

5.4.2 Supervision system

The main problems of the earlier system have been defined and could be summarized as follows: “We do not know in real-time what is happening on the production floor and when we finally get the information, it is too late to act”. A lot of time is wasted in finding faults because the chain of operations to do so is longer than the production time of most batches. Minimizing the chain of operations between detecting a problem and making a correction is the major task of the new supervision system. The target is to obtain a simplified approach to problem solving represented by the flowchart in Fig. 26. In order to achieve this level of simplification, some of the steps done earlier manually have to be automated.

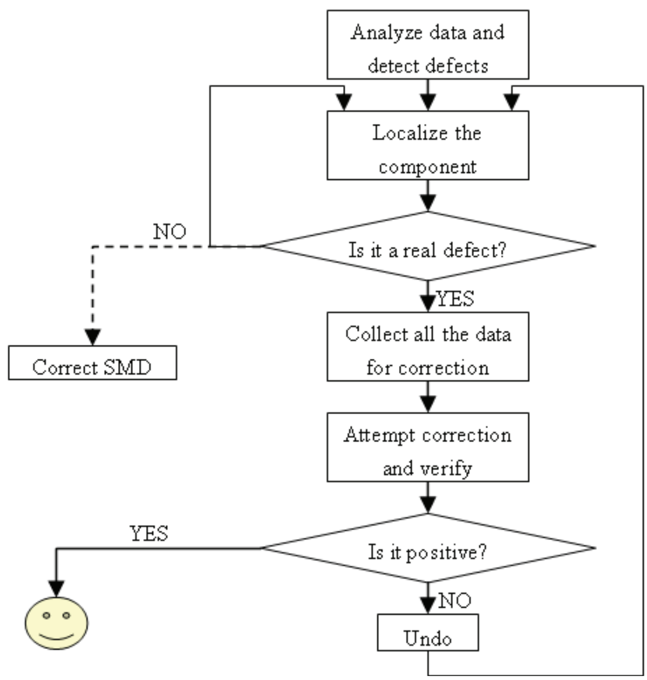


Fig. 26. Simplified problem solving approach of the new supervision system (Gebus & Ruusunen, 2004).

In order to minimize structural changes on existing systems, a high level functional approach has been used as shown in Fig. 27. The main functions of the supervision tool, their supporting systems as well as data that are being exchanged have been identified. This enables updating of existing software according to the real task that it should perform, the input data and the information generated as output. Each part of the software can be developed independently of other parts because as long as it fulfills its own sub-functions, it will necessarily fulfill the main objective of the supervision system.

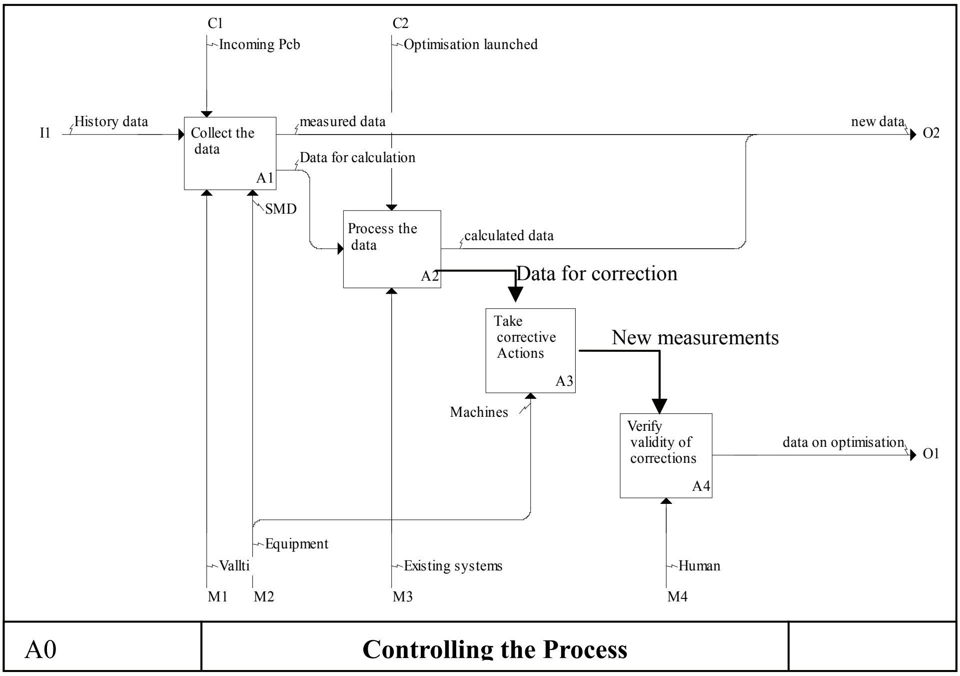


Fig. 27. SADT diagram of the new supervision system (Gebus & Ruusunen, 2004).

Control charts are a good way to avoid tampering with a process, but they do not provide information on how the process can be improved. It is therefore impossible to get the best performance out of a stable process. In order to determine a course of action, knowledge, whether it is acquired or generated, is more efficient than to blindly use statistical approaches.

In electronics manufacturing as well as in most manufacturing areas, production quality can be degraded by a shift of the mean or by an increase of the variance. In order to control the process, actions on these parameters have to be taken. In this case study, feedback information relative to the component misalignment is therefore needed. Furthermore, a simple and effective way with some key parameters easy to understand by line operators is required. The control system must be able to show precisely where a problem occurred, how big it is, and it must give information for the correction that has to be made. For these purposes four specific indices have been defined in order to characterize the behavior of the process over the last 20 measurements.

5.4.3 Quality indices and data analysis

Since the misalignment problem is symmetrical, only the details of calculation for X direction are given here. The following is the list of variables and parameters used for determining different indices. Most of these variables are standardized with the value 1 taken for the control threshold value of each individual measurement. This choice was made for dealing with different fault detection thresholds, but also for allowing generalization of the formulae proposed in this section.

- ***Dx_avg*:** Average sample shift on X direction (1/100mm).
- ***X_avg*:** Standard value of the average shift.
- ***X_max*:** Maximum value of the standard shift.
- ***X_max_refno*:** Reference number of the component causing *X_max*.
- ***X_min*:** Minimum value of the standard shift.
- ***X_min_refno*:** Reference number of the component causing *X_min*.
- ***k*:** Corrective factor used to take into account the sample size.
- a_1, a_2: Weighting coefficients used to calculate I_3.
- T_1, T_2: Warning and control limits for the indices.
- ***N*:** Number of history data available (usually 20 samples).

Defect detection Indices

Three indices have been designed especially for the problem of fault detection. I_1 calculated by:

$$I_1 = \frac{k}{N}\sum_{i=1}^{N} X_i\_avg \tag{16}$$

can be used to characterize sample shifts. If I_1 is high, it means that the component placement is steadily shifting toward the same direction. In normal conditions, the index should have a value between 0 and *k* since the process does not allow shifts to go beyond the threshold values without triggering an alarm and stopping production.

I_2 calculated by:

$$I_2 = k\frac{Avg(5biggestX\_\max) - Avg(5smallestX\_\min)}{2} \tag{17}$$

gives a qualitative indication on the variation of the sample values. This index has under normal conditions a value between 0 and *k*, and reaches high values when the process displays a significant variation.

I_3, which is defined by:

$$I_3 = a_1 I_1 + a_2 I_2 \tag{18}$$

is an additional index designed for taking into account the relative importance between shift and variation. Different weights can be set for defining a priority in the detection of shift and variation. However, this index is optional and no specific rule has been defined for setting these coefficient values.

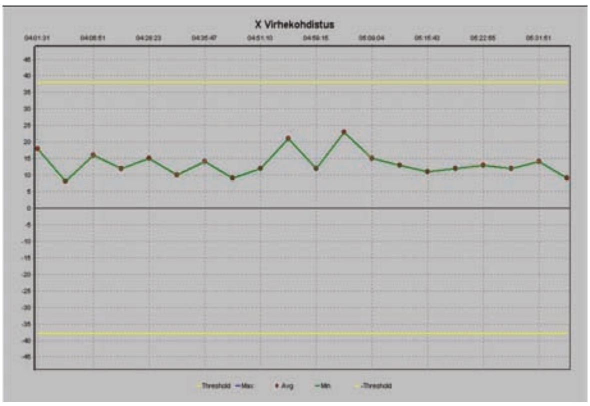
The rules for the different indices are:

- $I_1 < T_1$: No significant shift.
- $T_1 < I_1 < T_2$: Small shift and low priority (warning).
- $I_1 > T_2$: Big shift and high priority (problem).
- $I_2 < T_1$: No significant variation.
- $T_1 < I_2 < T_2$: Small variation and low priority (warning).
- $I_2 > T_2$: Big variation and high priority (problem).

The corrective factor is facultative but has been introduced for taking into account the amount of similar components on a PCB, whereas indices are calculated based only upon average, maximum, and minimum values. If an index is based on values from many components (*i.e.* the sample size is big), then it seems relevant to say that shift and variation are more critical, thus control limits should be tighter. But because control limits are fixed, introducing a corrective factor that increases with the sample size has a similar effect than tightening the control limits. In the case study, a linear function has been used for calculating the corrective factor.

Interpretation of data based on I_1 and I_2

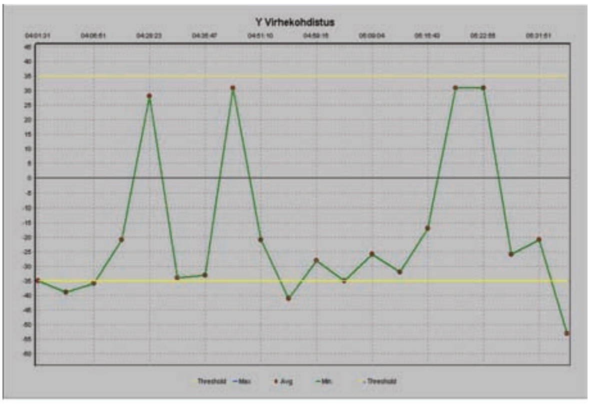
Fig. 28 to Fig. 32 show examples of data collected during the case study. Interpretation is based on the values of I_1 and I_2. It can be seen that the value of I_1 will indicate a process shift in all the cases, whereas I_2 is significant when the sample size is higher than 1. By providing information about the variation of individual component, it can potentially generate additional feedback information.



The component has first to be identified on the PCB in order to check if there is a real shift or only a measurement problem.

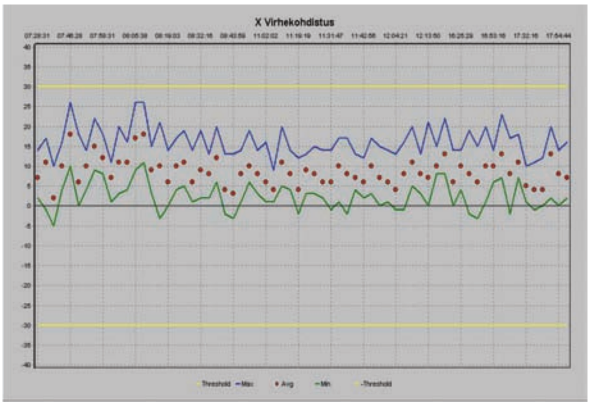
- The measurement problem can be solved by introducing an offset to the AOI system.
- For a real shift, corrections can be made on the placement machine.

Fig. 28. Case 1 - Interpretation of data with sample size 1, I_1 medium, and I_2 small.



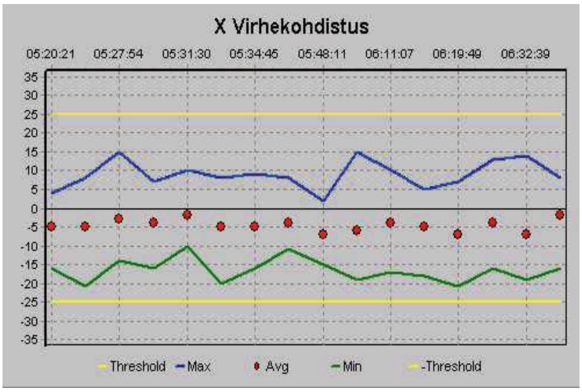
Before trying to make any correction related to the shift, the reason behind the variation has to be found.

Fig. 29. Case 2 - Interpretation of data with sample size 1, I_1 medium, and I_2 big.

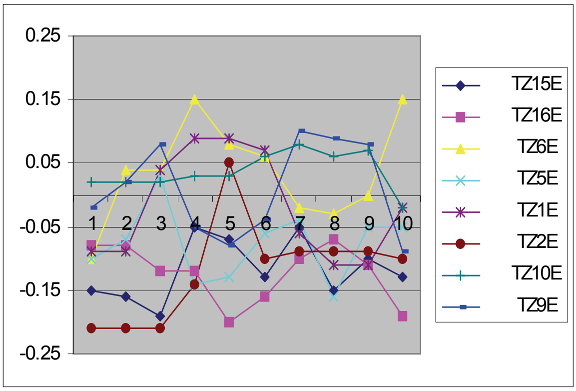


This case can be handled in the same way as case 1. Real shifts have to be separated from measurement problems before taking corrective actions.

Fig. 30. Case 3 - Interpretation of data with a sample size > 1, I_1 medium, and I_2 small.

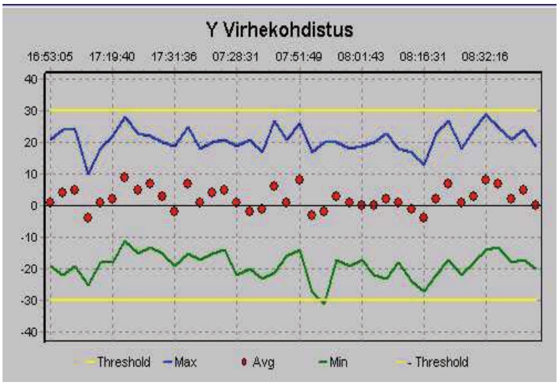


This case has to be checked in closer details to have the information on individual components.

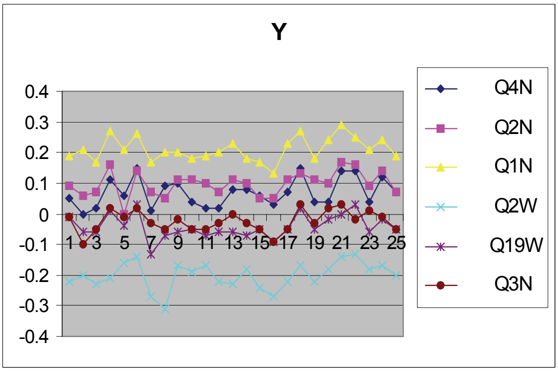


These details show significant variation for each individual component and almost each curve has a different shape. This is the most complicated case and the solution is difficult to find. Focusing on this kind of problem makes only sense if the batch size is important enough.

Fig. 31. Case 4 - Interpretation of data with a sample size > 1, I_1 small, and I_2 medium.



Like case 4, details of individual components have to be checked. However, it is already possible to notice that the Min. and Max. curves have similar shapes.



Individual components display similar shapes and only a very small variation. It is therefore possible to apply case 1 to each shifted component, mainly Q1N and Q2W in order to reduce variation of the whole sample.

Fig. 32. Case 5 - Interpretation of data with a sample size > 1, I_1 small, and I_2 big.

Feedback control index

As illustrated by the previous examples, not all cases will allow simple corrective actions to be taken. The case study focused on problems that would enable the operator either to realign a single component by adding an offset on the associated placement machine, or then to tune the AOI system if it turns out that a false alarm was generated. In order to determine if this form of feedback control is possible, a further index I_4 has been defined as follows:

$$I_4 = k.\underset{refno}{Max}\left|\sum_{i=1}^{N}\frac{X_i\_\max\_refno - X_i\_\min\_refno}{N}\right| \qquad (19)$$

with *refno* being the unique reference number of each individual component. $X_i\_max\_refno$ = 1 if that component is responsible for the maximum value of sample i and 0 otherwise. In a similar way, $X_i\_min\_refno$ = 1 if the component is responsible for the minimum value of sample i and 0 otherwise.

Practically, by applying formula (19), one can count how many times a single component has been responsible for the maximum value of each sample minus how many times it has been responsible for the minimum value. The resulting index represents the impact of a certain component on the fault, thus it can show the shift of the single component among many others.

The rules of interpretation for I_4 are as follows:

- If a component among the sample is severely and steadily shifted at a maximum or minimum, then the index will be high.
- If the sample size is 1 then *X_max_refno* - *X_min_refno* will always be null, thus I_1 and I_2 provide sufficient information.
- If the sample size is 2 then each component can only be max or min. Therefore, I_4 will only be high if both plotted curves have similar shapes, avoiding false alarms.
- If the same component is responsible for both maximum and minimum values, then the index will be low. This case is out of the scope for this index.

Calculation of a corrective action

As defined earlier, corrective actions take the form of offset changes that are applied for specific components on placement machines. In case of a sample size of only one component, the correction is equivalent to the average shift *Dx_avg* and is defined by:

$$Correction = \pm Dx\_avg \tag{20}$$

In a similar way, in case of a sample size of more than one component but with a small variation, *X_min*, *X_max* and *X_avg* can be assimilated to a unique value. Correction is therefore again given by (20), but this time applied independently to all the components in that sample.

In case of a sample size of more than one component but with a small number of components causing high variation in the data, I_4 will be high. Nevertheless, the correction will still be the same. The only difference is that it is only applied to the incriminated component thus reducing the variation. Table 2 gives an example of corrective action resulting in a lower variation of data. In this case, the sample size was three but component IC7G alone was responsible for 87.5% of all maximum values. This situation is detected by index I_4 and corrective action can be taken utilizing equation (20) for the specific component.

Table 2. Example of results before and after corrective actions.

	Y_avg before correction	Y_avg after correction	
I_1	0.06	0.04	No global shift
I_2	*0.413*	0.298	Variation on Y reduced
max_refno	IC7G	IC7G	
	87.5%	*75%*	
min_refno	IC1CM	IC1CM	
	62,5%	62,5%	
I_4	*89.3%*	*76,5%*	Variation on Y due to IC7G

5.4.4 Implementation and results

Software was already available on the production floor for monitoring quality parameters or displaying product pictures. However, these were independent and not based on relevant data. Implementation of the method consisted therefore mainly of updating existing software so that it could perform the function it was originally meant for. Major improvements were done in that way. Data availability was addressed in handling of databases, data compression techniques were improved, monitoring and analyzing functions were included, and finally, locating the faulty components was done by integrating a user interface to the supervision tool.

Concerning the database, all the data necessary for controlling the process is located in a unique table, and only the necessary data is in that table. The problem of data availability has been solved by systematically saving the 20 last values, whenever they have been produced. Despite not being very reliable, old data is nevertheless better than no data at all.

Existing supervision interface fulfilled none of the given targets, as it did not provide any feedback information that would allow corrective actions to be taken. Therefore, two specific functions have been added besides graphical improvements as shown in Fig. 33.

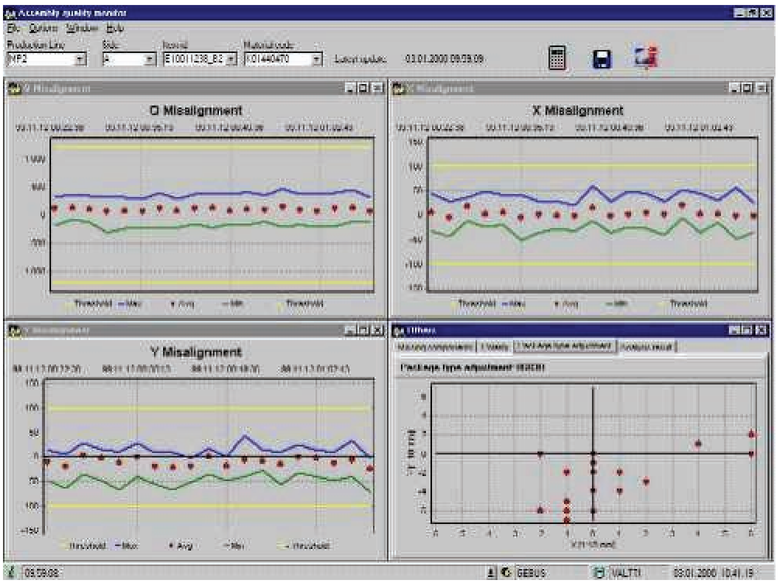


Fig. 33. Main window of the supervision interface (Gebus & Ruusunen, 2004).

First, a browsing capability has been added to include a monitoring function. For each selected quality index, it is now possible to get a list of material codes sorted out from the highest to the lowest probability of the defect. In case of recurrent problems, it is possible to navigate in a list of material codes according to a specific defect.

Data analysis capabilities constitute the second additional function. When a material code has been selected, details of the analysis can be accessed and warnings are given for values higher than predefined thresholds as shown by the example in Fig. 34. It can be seen that there are ten similar components for this material code, but 89 percent of the maximum or minimum values are due to the same component R35. This information means that the component R35 is at least partly responsible for the 48 percent high variation. A correction equal to the average shift of -0,03mm can therefore be taken on this particular component by changing the relevant offsets of the placement machine.

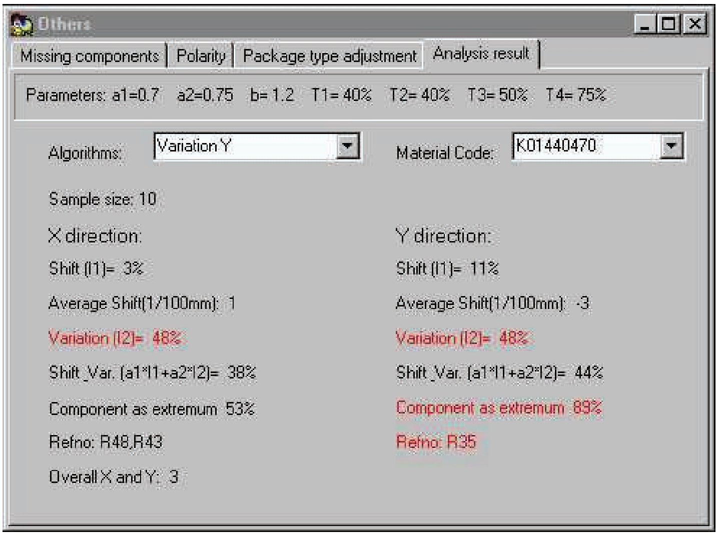


Fig. 34. Data analysis window of the supervision interface (Gebus & Ruusunen, 2004).

An output interface showing a picture of the PCB with missing components already existed but here updates relative to feedback control have also been made. Earlier it was not possible to locate a specific component on a board, unless the complete information for the board, side and component, etc were known. This information was scattered and not easily accessible. Modifications and a link have been made with the supervision interface. When a material code is selected from the supervision software, then the output interface automatically shows the right picture and the list of components for the selected material code. When one component is selected, the picture is automatically centered on that component as shown in Fig. 35. This allows easy and fast localization of components on the PCB, which enables operators to verify if a defect is real or only a false alarm generated by a measurement problem.

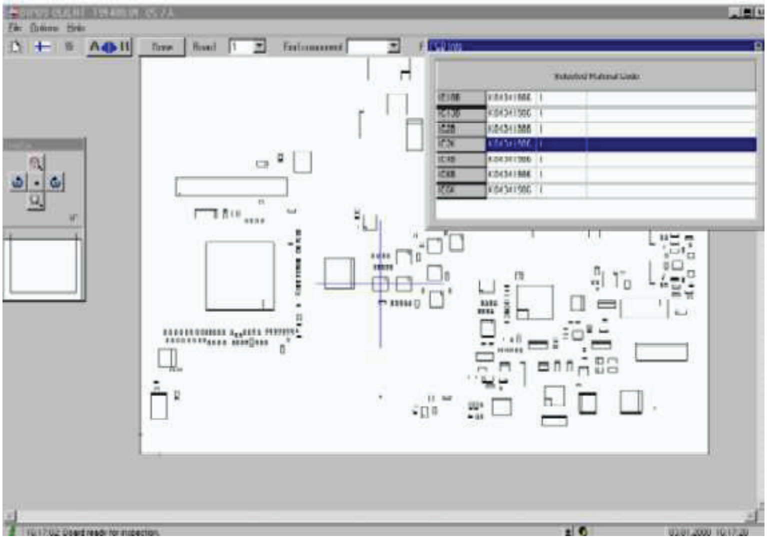


Fig. 35. Defect localization window of the supervision system (Gebus & Ruusunen, 2004).

At the end of this case study, the system had been implemented and running on-line for a period of six months. Although this is not a long test period, the results were promising.

The methods used need a minimum amount of data in order to be efficient and produce reliable information. Results may be better with a heavy statistical analysis, but from a quality point of view the system manages to provide short-term feedback with a small amount of data. While having had to walk up to 400 meters between the detection of a defect and an outdated corrective action, operators now can obtain all the necessary information automatically and walk the 10 meters separating them from the farthest placement machine.

Using the system on-line also produced a large amount of false alarms. The reason for this is inherent to the type of optical control, based on pattern learning. False alarms are in fact due to the measurement reliability and not the products themselves. It has been possible to tune up the control system and thereby reduce false alarms. This is especially the case when using the system for prototype production.

Furthermore, the supervision system has been designed in an open way. It fits to the production configuration at this specific production plant, but can easily be adapted to other kinds of production lines. Usually, processes in the electronics industry are closer to mass production, and for those cases, statistical features and additional tests should be added. However, the structure of the program makes these kinds of upgrades easy to implement.

The new supervision system also had a beneficial effect on line operators. By simplifying their tasks, it made it possible for them to have an active role in terms of quality feedback, thus improving their understanding of the process and its problems. Interfacing the controllers of an AOI station with a repair station allows the transmission of a defects list of the device under test to the repair station. At the repair station efficient defect information can enable quick response and repair actions by the operator. At this stage also verification or falsification of the testing results of the automated inspection system may be done. This enables the further optimization of the parameter settings at the AOI system in order to reduce its escape and false alarm call rate.

However, since the method already provides a quantitative response to the defect, one could easily imagine an automatic control of the process. Further research could also focus on measurement validation and automatic tuning of the monitoring system in case of false alarms.

5.5 Conclusion

Statistical process control is a tool primarily aimed at continuously monitoring the common cause system and detecting significant deviations, possibly pointing to special assignable causes. It appropriately employs considerations of statistical significance to trigger action and so reduces the chance of fruitless pursuit of phenomena produced by chance alone.

The use of SPC is no panacea especially when used alone. The first problem derives from the fact that SPC does not allow tuning of the system while it is operating because the statistics rely on the stability of the process conditions. This limitation makes it impossible to have a system running at an optimal level. Therefore, more advanced Run-to-Run techniques have been developed to provide faster feedback control. Secondly, the blind use of statistical methods only underlines a lack of expertise in the domain problem. Statistical methods should in fact be used only when knowledge of the situation is lacking and cannot be obtained. If direct knowledge about a situation is available, then statistical probabilities should not be blindly followed. In other words, if some form of knowledge about a situation is available, action should be based upon it.

The approach presented in the case study tries to propose solutions to these problems. A supervision system using an approach based on methods derived from traditional SPC has been adapted and implemented in an environment poor in data, typically small batches or prototype production. Furthermore, available data is incomplete and compressed leading to the development of a new type of control chart. Subsequent control schemes are based upon the calculation of a few key parameters for generating new defect information, which makes fault diagnosis possible. Despite its simplicity, the method is very efficient and is easy to implement as part of a larger supervision system.

The supervision system also raised the interest of operators on the topic of quality. By providing them with adequate information to adjust the process when needed, it made them active players in quality management, which is essential in any quality improvement actions.

6 Discussion

6.1 Integration of data and knowledge

The different examples of this thesis and especially case study No.1 demonstrate the practical feasibility of automatic or semiautomatic integration of knowledge along with existing data-based approaches and within the larger context of decision support systems. The hypothesis of this thesis was that a better integration of knowledge leads not only to improved fault detection and recovery, but also to a better defect-related communication.

For the first point, major improvements concern the efficiency at the production floor of diagnosis and maintenance operations. This efficiency can be measured in several ways. It has been shown for example that the use of human expertise as a source of knowledge can compensate for a lack of relevant data. This leads to an improvement both in accuracy and speed of response compared with traditional methods as knowledge-based approaches rely less on historical data. In cases when data is available but contains mistakes (case study No.3), the system is able to detect the problem.

Usually knowledge is considered as context-dependent. For this reason, companies are sometimes reluctant to invest in the development of solutions that only have a local impact. It has been demonstrated however that a proper conceptualization of knowledge allows an acceptable level of flexibility. Although this adaptability requires a bigger involvement of experts at the programming stage, this is largely compensated by the decrease in the necessary manpower to operate such knowledge-based systems. For example, expertise that had to be acquired by each repair worker before it becomes operational can be encompassed once and for all within the decision support system.

Concerning defect-related communication, improvements linked to the integration of knowledge can be seen mainly at two levels. First, through the proper choice of interfaces and knowledge conceptualization, knowledge-based systems can be tailored to the user resulting in a working two-way communication. The decision system can be made to speak an easily understandable language providing only the information needed by the user. A consequence of this, as has been noticed in all the case studies, is a raised interest and involvement from operators toward quality. They are more eager to use a knowledge-based system, thus making their expertise available. This brings about a second major improvement concerning defect-related communication as this expertise is not only stored within the system but also shared among the users. Knowledge is therefore not anymore restricted to a single user or even to the factory floor.

6.2 Limitations of the different methods developed in this thesis

In an attempt to create an advisory system for the specific needs of the electronics industry, different aspects of a decision support system have been studied in this thesis. For such a system to be efficient, it would have to perform the following series of tasks within a specific environment:

- The system must help collecting data, information, and knowledge about the manufacturing conditions of a product.
- It must analyze what has been collected and reach automatically conclusions that would otherwise require the intervention of quality control and process specialists.
- Finally, it must communicate these conclusions in an intelligible way to the relevant people when needed.

This brings about several issues that the methods proposed in this thesis try to solve.

6.2.1 Obtaining meaningful inputs for the DSS

Feigenbaum & McCorduck (1983) define knowledge acquisition as a major bottleneck when building advisory systems relying on human expertise. Useful features for analyzing process conditions are often expressed in qualitative terms, which is not the most adequate form for computational processing. It is, however, the form in which the people with the ability to identify those features express their knowledge.

Collecting and interpreting this knowledge is recognized to be a very difficult task by any expert from the field of cognitive engineering. Problems such as knowledge bias, its availability, or the unwillingness of the expert to relinquish what he considers power are just some of the problems that might be encountered. 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 focuses on rather narrow areas of the problem. Kawaguchi *et al.* (1991) tried to automate the interview process, Okamura *et al.* (1991) focused their efforts on shallow knowledge, Tecuci (1992) and Winter (1992) dealt with the problem of incomplete or faulty knowledge bases, whereas Chien & Ho (1992) defined generic knowledge categories. But from the author's knowledge about the topic, generalization of knowledge acquisition approaches in an attempt to create generic tools has mostly resulted in tools that are poorly fitted for any specific case.

All three case studies in this thesis deal at some level with the problem of acquiring adequate knowledge for solving specific problems. Although it was not thought to be an important part of this research at first, it soon became clear that acquiring knowledge was essential in achieving any of the initial targets. Whether manual methods or automatic acquisition systems were used, the approach had to be human-oriented in the end. It does not only have the potential to improve data analysis, but it also increases employees' awareness toward quality issues.

Case study No. 3 came first in chronological order. Here, a skilled technician who teaches the AOI system good component patterns provides knowledge. He also tunes the system in case of false alarms. The problem is that this technician needs specific training to fulfill his tasks.

Case study No. 2 came second. It saw the arrival of an interface designed specifically for knowledge acquisition. Design engineers at one end of the company feed a defect localization system with useful information, so that the repair engineers at the other end of the manufacturing chain can benefit from an understandable diagnostic. It was already a major improvement from Case study No.3, but still some problems persisted. The biggest problem of all, however, was the lack of the feedback loop from repair to design leading to poor motivation from the latter one into fulfilling its tasks properly.

Case study No.1 is an attempt to solve some of the problems encountered previously by transforming knowledge acquisition into a non-intrusive multi-level information sharing system as in Fig. 36 that benefits directly to all its users. Some issues however remain:

- Although dedicated interfaces were developed for users with various degrees of knowledge, these cannot be considered to be fully flexible interfaces, as this terminology refers usually to the kind of interfaces that learn from their users and adapt to their individual needs. In the case studies, however, such a level of flexibility was not required.
- As far as the author is aware, 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 proposal was made to increase flexibility and portability of the system. However, this might also limit the ability to catch the full scope of information as two pictures of each cell might be insufficient to locate all possible faults.
- Digital photography aims at simplifying the knowledge extraction process by creating an environment common and, most of all, familiar to all users. Fault description, however, does not benefit from this standardization, which means that new problems and misunderstandings can arise on this side.

When trying to implement a KBS, one additional concern is to provide it with the ability to communicate efficiently and transparently within a factory environment. The approach given in this thesis proposes to focus on three user-oriented axes of improvement: ***Usability***, ***Usefulness***, and ***Usage*** (3U). These, however, are more directions of improvement as it is very difficult in a context specific environment to provide precise general answers.

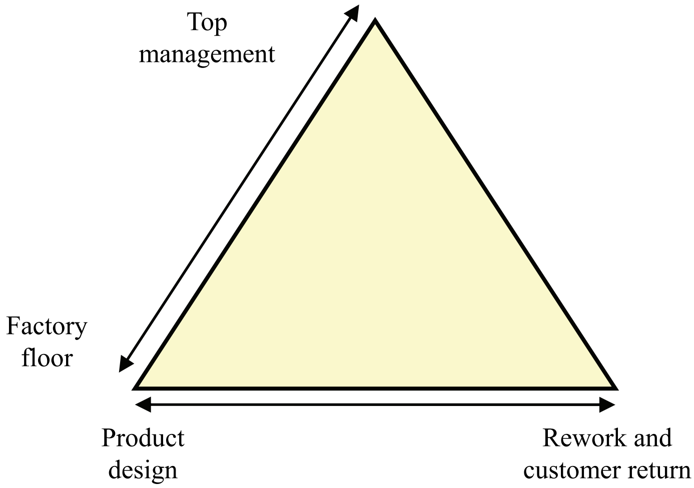


Fig. 36. Area of application of a multi-level information sharing system.

6.2.2 Analyzing process conditions

As the complexity of electronic products increases, diagnosing the defects has become of paramount difficulty. Running complete test sequences is known to be a NP-complete problem. This problem is sometimes broken down into more specific tasks such as detection, localization, and isolation of a defect.

Over time, many methods besides brute force were developed to deal with diagnosis. These include data analysis, rule-based systems, model-based reasoning, and case-based reasoning. But many of these methods do not make the most efficient use of qualitative data for analyzing defective situations. This data, however, often encompasses the information for properly diagnosing a situation and proposing corrective actions for fault recovery. Hybrid systems such as fuzzy expert systems or fuzzy case-based systems attempt to provide solutions to handling qualitative information.

The author's contribution to this topic, described in Case study No.2, uses a linguistic equation approach to create a rule base for an expert system. This rule base is entirely generated from information obtained from design engineers in a qualitative form. It can therefore be considered to be a way of conceptualizing expert knowledge into a form that is more suitable for computational processing. The system developed by this method is able to provide useful diagnostic with only very few historical training data. Some issues, however, remain to be solved:

– Although the method provides a robust and reliable approach to defect detection, it is currently still impossible to isolate these defects, which means that it is not possible to separate different defects. This limitation originates from the fact that the rule base

only uses expert knowledge, which is in this case insufficient for defect identification. It is therefore only possible by now to separate defective products from faultless ones.
- Relying more on experimental data would require a much larger amount of training data, which means data from clearly identified defects. This could lead to a much narrower defect localization. However, a sufficient amount of data would only be available after several months of production, which is not acceptable in an environment where production systems tend to evolve almost every year.

The approach maybe lacks in precision concerning defect identification. Nevertheless, it has proven to be efficient in detecting and locating defect areas with training data representing much less than a day of production.

6.2.3 Real-time issues

A problem that many control strategies face is to provide feedback within a short enough time to enable corrective actions and prevent the manufacturing of faulty products. This is a major drawback especially for SPC-based inspection strategies as they depend on the stability of the process. Hence, they do not allow real-time tempering with the process parameters.

Run-by-Run control tries to elude the basic stability assumption requested by the traditional SPC approach. Information from previous runs is used to tweak process parameters in an attempt to continually improve the outputs. A good RbR controller should be able to compensate for various disturbances through the use of both gradual and rapid modes.

The method suggested in Case study No. 3 tries to complement traditional SPC procedures in the case of a lack of relevant data through the use of a few representative quality indices. Although it cannot be considered as real-time control, this approach can be applied to prototype production lines and more generally any process requiring short-term feedback. Some issues can still arise when using the quality indices:

- The method is not suitable for multivariate approaches that nevertheless represent a current trend in the industry as many problems can only be detected through combined effects on multiple variables. However, as far as component placement is concerned, the method is well suited as this is subject to structural testing by AOI systems and hence, it remains a univariate problem.
- Although the system suggests a corrective action for a certain component, this suggestion cannot be fully trusted as many false alarms are generated. The reason is that the quantified suggestion originates from the measured component shift, which in turn depends very much on the correctness of the learning pattern used to train the AOI system. Hence, unless a large amount of accurate historical data is available for properly tuning the inspection system, the proposed corrective action will be inaccurate.
- The application domain of the quality indices is restricted to shifted processes but cannot handle out-of-control processes.

6.3 Ideas for further research

The research over the past years has focused on solving some of the issues encountered when trying to implement knowledge-based decision support systems in an electronic manufacturing environment. The author tried to cover a wide range of problems from acquiring knowledge that might improve the assessment of an industrial situation, to communicating the result of that assessment to the relevant people. Nevertheless, some points might require further research, especially concerning the current limitations described in section 6.2. Furthermore, in a general sense, both the inputs and outputs of DSS can still be improved for further widening the range of problems that are included.

System inputs can be improved by machine learning techniques, thus further simplifying the knowledge acquisition process. The aim is to make any knowledge-based system as transparent as possible to the users. Solutions proposed in this thesis try to be simple to use and, for the latest versions, interesting for all the users. Nevertheless, they all still require the user's commitment to feed the system with data. An ideal system should be able to generate its own knowledge only from observing and interpreting its environment.

The next step for system output is to develop automatic feedback control, which means that corrective actions are taking place without human intervention. This seems, under current circumstances, more of a utopia as it necessitates standard and well-known production tools. It is, however, difficult to imagine automatic control, which is very specific to the tools and technologies used for manufacturing, in an ever-evolving environment.

7 Conclusion

Most existing process control systems have been developed for the mass production industry and are therefore suited to that kind of production. When confronted by an evolving environment these systems are no longer reliable. In electronics manufacturing, constant innovation in products and production systems makes it hard to gain enough knowledge for using traditional production monitoring methods. New ways of combining traditional knowledge-based methods and intelligent data analysis are therefore needed. One difficulty is to obtain applications that are not computationally demanding. One of the main constraints is to have a real-time system using reactive methods.

In this thesis, different aspects of a knowledge-based decision support system have been presented within the specific context of the electronics industry. More specifically, the dissertation addresses three main topics at different stage of the design of the knowledge-based decision support system. The first stage concerns knowledge acquisition, often considered as the bottleneck in the design process of a knowledge-based system. Unlike traditional design techniques that emphasize on doing right the first time, the 3U approach proposed in this thesis leads to a better match with user concern Knowledge extraction methods have been investigated in order to adapt acquisition interfaces to different levels of knowledge, as well as different knowledge sources. At the second stage, combination of knowledge and data enables performance improvements of traditional data mining techniques. To do so, knowledge elicitation methods were developed based on fuzzy logic and linguistic equations that make the integration of data and knowledge possible. The final stage concerns real-time feedback control. The interest is to study the implementation of knowledge-based data analyzing methods, and their ability to produce explicit diagnosis. In order to increase the lifetime of such methods, self-tuning capabilities were also investigated. A new type of control chart based on three fault indices and applicable to incomplete and compressed data has been developed.

The research results show that it is possible to develop new kinds of decision support systems through tools and methods that improve the integration of knowledge in an evolving environment. Moreover, they also emphasize the essential need for a real commitment of management structures if any of the proposed solutions has to be viable after the beta implementation.

References

Balakrishnan A & Semmelbauer T (1999) Circuit diagnosis support system for electronics assembly operations. Decision Support Systems 25(4): 251–269.

Balasubramanian V, Turoff M, & Ullman D (1998) A Systematic Approach to Support the Idea Generation Phase of the User Interface Design Process. In: 31st Annual Hawaii International Conference on System Sciences, vol 6: 425–434.

Boning D, Moyne W, Smith T, Moyne J, & Hurwitz A (1995) Practical issues in run by run process control. In: IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop: 201–208.

Boyd JD & Banan M (1995) Run by run (generalized SPC) control of semiconductor processes on the production floor. In: 17th IEEE/CMPT International Electronics Manufacturing Technology Symposium: 60–64.

Bálint L (1995) Adaptive Interfaces for Human-Computer Interaction: A Colorful Spectrum of Present and Future Options. In: IEEE International Conference on Systems, Man and Cybernetics, vol 1: 292–297.

Del Castillo E & Rajagopal R (2002) A multivariate Double EWMA Process Adjustment Scheme for Drifting Processes. IIE Transactions 34: 1055–1068.

Champagne M & Monette R (2002) Batch multivariate SPC monitoring of a sulfite pulp digester. In: 2002 American Control Conference, vol 1: 606–607.

Chaudhry N, Moyne J, & Rundensteiner E (1996) Utilizing Fuzzy Data and Rules for Discrete Control of Semiconductor Manufacturing Processes. In: 5th International Conference on Data and Knowledge Systems for Manufacturing and Engineering: 199–222.

Chen A, Guo RS, & Yeh PJ (2000) An effective SPC approach to monitoring semiconductor manufacturing processes with multiple variation sources. In: 9th International Symposium on Semiconductor Manufacturing: 446–449.

Chien CC & Ho CS (1992) Design of a Generic Knowledge Acquisition Shell. In: IEEE International Conference on Tools with AI: 346–349.

Crowder SV (1989) Design of exponentially weighted moving average schemes. Journal of Quality Technology 21: 155–168.

Cullen CC & Bothe DR (1989) SPC for short production runs. In: IEEE National Aerospace and Electronics Conference, vol 4: 1960–1963.

Cuéllar J (1991) Statistical process control for nonnormally distributed variables through the use of transformations. In: IEEE/SEMI Advanced Semiconductor Manufacturing Conference: 143–148.

Dixon NM (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. In: IEEE International Conference on Systems, Man, and Cybernetics. vol 2: 1331–1335.

El-Sayed MA & Alfuhaid AS (2000) ANN-based approach for fast fault diagnosis and alarm handling of power systems. In: 5th International Conference on Advances in Power System Control, Operation and Management: 54–58.

El-Shal SM & Morris AS (1999) A fuzzy rule-based algorithm to improve the performance of SPC in quality systems. In: IEEE International Conference on Systems, Man, and Cybernetics, vol 5: 284–289.

El-Shal SM & Morris AS (2000) A fuzzy expert system for fault detection in statistical process control of industrial processes. IEEE Transactions on Systems, Man and Cybernetics C30(2): 281–289.

Fan CM, Guo RS, & Chang SC (1997) EWMA/SD: an End-of-Line SPC Scheme to Monitor Sequence-Disordered Data. In: 2nd International Workshop on Statistical Metrology 8: 66–69.

Fan CM, Guo RS, & Chang SC (2000) SHEWMA: an end-of-line SPC scheme using wafer acceptance test data. IEEE Transactions on Semiconductor Manufacturing 13(3): 344–358.

Feigenbaum EA & McCorduck P (1983) The fifth generation: artificial intelligence and Japan's computer challenge to the world. Addison-Wesley Longman Publishing Co.

Feldmann K & Sturm J (1993) Yield Improvement in SMT production by integrating Process Monitoring and Testing. In: IEEE/CHMT International Electronics Manufacturing Technology Symposium: 229–234.

Feldmann K (2003) Challenges in Electronic Production. Chapter 13 of Manufacturing Technologies for Machines of the Future: 21st Century Technologies by Anatoli I. (EDT) Dashchenko: 381–412.

Fujihara H, Simmons DB, Ellis NC, & Shannon RE (1997) Knowledge Conceptualization Tool. IEEE Transactions on Knowledge and Data Engineering 9(2): 209–220.

Gaines BR (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 (2000) Process Control Tool for a Production Line at Nokia. Oulu University Press.

Gebus S & Juuso EK (2002a) Defect Detection on Printed Circuit Boards using Linguistic Equations applied to Small Batch Processes. In: eunite 2002 symposium: 545–550.

Gebus S & Juuso EK (2002b) Industrial utilization of Linguistic Equations for Defect Detection on Printed Circuit Boards. In: IECON'02 IEEE conference: 1887–1892.

Gebus S & Juuso EK (2003) Expert System for Defect Localization on Printed Circuit Boards. In: AUTOMAATIO’03 Seminar Days: 425–430.

Gebus S & Ruusunen M (2004) Decision Support System for Short Term Statistical Feedback Control on a Semiconductor Process. In: IEEE International Symposium on Industrial Electronics: 247–252.

Gebus S, Soulas A, Fournier G, & Ruusunen M (2004) Design of an Interface for Knowledge Acquisition on the Production Floor. In: eunite 2004 symposium: 139–144.

Gebus S, Fournier G, Vittoz C, & Ruusunen M (2006) Knowledge Extraction for optimizing Monitorability and controllability on a Production Line. Oulu University Press.

Goel AL & Wu SM (1973) Economically optimum design of CUSUM charts. Management Science 19: 1271–1282.

Ha S, Sachs E, & Ingolfsson A (1990) Run by Run Process Control. In: 9th IEEE/CHMT Electronic Manufacturing Technology Symposium, 'Competitive Manufacturing for the Next Decade'.

He J, Xu J, & Yao X (2000) Solving Equations by Hybrid Evolutionary Computation Techniques. IEEE Transactions on Evolutionary Computation 4(3): 295–304.

Holsapple CW, Park S, Stansifer RD, & Whinstn AB (1988) Flexible user interface for decision support systems. In: 31st Annual Hawaii International Conference on System Sciences, vol 3: 217–222.

Holsapple CW, Tam KY, & Whinston AB (1989) Building Knowledge Acquisition Systems – A Conceptual Framework. In: 22nd Annual Hawaii International Conference on System Sciences, vol 3: 200–210.

Hu A, Sachs E, & Ingolson A (1992) Run-by-Run Process Control: Performance Benchmarks. In: IEEE/SEMI International Semiconductor Manufacturing Science Symposium: 73–78.

Hyötyniemi H (2001) Multivariate Regression – Techniques and Tools. Helsinki University of Technology, Control Engineering Laboratory.

IGL Technologie (1989) SADT un langage pour communiquer. Paris, Eyrolles.

Isermann R & Ulieru M (1993) Integrated Fault Detection and Diagnosis. In: International Conference on Systems, Man and Cybernetics, vol 1: 743–748.

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

Juuso EK (1994) Fault Diagnosis based on Linguistic Equation Framework. In: IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes: 374–379.

Juuso EK, Ahola T, & Oinonen K (1998) Case-Based Reasoning with Linguistic Equations for Web Break Sensitivity Indicator. In: Tool Environments and Development Methods for Intelligent Systems Symposium: 97–106.

Juuso EK (1999) Fuzzy Control in Process Industry: The Linguistic Equation Approach. In: Fuzzy Algorithms for control. International Series in Intelligent Technologies: 243–300.

Juuso EK (2004) Integration of intelligent systems in development of smart adaptive systems. International Journal of Approximate Reasoning 35: 307–337.

Kang BS & Park SC (2000) Integrated machine learning approaches for complementing statistical process control procedures. Decision Support Systems 29: 59–72.

Kawaguchi A, Motoda H, & Mizoguchi HR (1991) Interview-Based Knowledge Acquisition Using Dynamic Analysis. IEEE Expert: 47–60.

Keiner W (1990) A Navy Approach to Integrated Diagnostics. In: Autotest Conference Record. IEEE Press: 443–450.

Kokott J (2003) Solder joint inspection in high mix/low volume – advancing with fewer cameras. GOEPEL Electronic, EPP Europe 3-4/2003.

Komulainen K, Heikkinen M, Frantti T, Juuso EK, & Leiviskä, K (1997) Utilization of failure information in functional testing. In: Tool Environments and Development Methods for Intelligent Systems Symposium: 109–115.

Kuo RJ, Kuo YP, & Chen KY (2005) Developing a diagnostic system through integration of fuzzy case-based reasoning and fuzzy ant colony system. Expert Systems with Applications 28: 783–797.

Lambert SC & Ringland GA (1990) Knowledge Representation and Interfaces in Financial Expert Systems. UK IT Conference: 434–441.

Liebowitz J (1993) Educating Knowledge Engineers on Knowledge Acquisition. In: IEEE International Conference on Developing and Managing Intelligent System Projects: 110–117.

Lind S & Marshak W (1994) Cognitive Engineering Computer Interfaces: Part 1 – Knowledge Acquisition in the Design Process. In: IEEE National Aerospace and Electronics Conference, vol 2: 753–755.

MacDonald BA & Witten IH (1989) A Framework for Knowledge Acquisition through Techniques of Concept Learning. IEEE Transactions on Systems, Man, and Cybernetics 19(3): 499–512.

MacGregor JF & Kourti T (1995) Statistical Process Control of Multivariate Processes. Control Engineering Practice 3: 403–414.

Marchand H, Boivineau O, & Lafortune S (2001) Optimal Control of Discrete Event Systems under Partial Observation. In: 40th IEEE Conference on Decision and Control, vol 3: 2335–2340.

Margavio TM, Conerly MD, Woodall WH, & Drake LG (2004) Alarm rates for quality control charts. Statistics & Probability Letters 24: 219–224.

Marshak W & Lind S (1994) Cognitive Engineering Computer Interfaces: Part 1 – An Objective Design Process. In: IEEE National Aerospace and Electronics Conference, vol 2: 756–760.

Martin EB & Morris AJ (1995) Multivariate statistics and neural networks in process fault detection. IEE Colloquium on Qualitative and Quantitative Modeling Methods for Fault Diagnosis: 7/1–7/8.

Martin EB, Morris AJ, & Zhang J (1996) Process performance monitoring using multivariate statistical process control. Control Theory and Applications 143 (2): 132–144.

McGraw KL (1994) Knowledge Acquisition and Interface Design. Software, IEEE 11(6): 90–92.

McKeon A & Wakeling A (1989) Fault Diagnosis in analogue circuits using AI techniques. In: International Test Conference. IEEE CS Press: 118–123.

Morgenstern VM, Upadhyaya BR, & Benedetti M (1988) Signal anomaly detection using modified CUSUM method. In: 27th IEEE Conference on Decision and Control, vol 3: 2340–2341.

Moyne J, Del Castillo E, & Hurwitz AM (2000) Run-to-Run Control in Semiconductor Manufacturing. CRC Press.

Nakatsu RT & 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 AF & Stanley J (1989) Adaptive Human-Computer Interfaces: A Literature Survey and Perspective. IEEE Transactions on Systems, Man, and cybernetics 19(2): 399–408.

Nugent BA (1990) SPC in short run processes. In: IEEE National Aerospace and Electronics Conference, vol 3: 1304–1307.

Oakland JS & Followell RF (1990) Statistical Process Control: A practical guide. Oxford: Heinemann Newnes Editor, second ed.

Okamura I, Baba K, Takahashi T, & Shiomi H (1991) Development of Automatic Knowledge-Acquisition Expert System. In: International Conference on Industrial Electronics, Control and Instrumentation, vol 1: 37–41.

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. Page ES (1954) Continuous inspection schemes. Biometrika 41: 100–115.

Pan R & Del Castillo E (2001) Comparison of some process adjustment methods and their integration with SPC charts. In: INFORMS 2001, Miami.

Pan SL & Scarbrough H (1999) Knowledge Management in Practice: An Exploratory Case Study. Technology Analysis and Strategic Management 11(3).

Paoli A & Lafortune S (2003) Safe Diagnosability on Discrete-Event Systems. Technical Report No. CGR-03-02, System Science and Engineering Division, Department of Electrical Engineering and Computer Science, The University of Michigan.

Pateras S & McHugh P (1997) BIST: A Test & Diagnosis Methodology for Complex, High Reliability Electronics Systems. In: IEEE Autotest conference: 398–402.

Pierce JA, Rodi LL, & Dalton RE (1988) Interactive Knowledge Acquisition for Fault Isolation and Repair. In: 4^{th} Conference on Artificial Intelligence Applications: 421.

Polanyi M. (1966) The Tacit Dimension. London: Routledge & Kegan Paul.

Power DJ (2004) Free Decision Support Systems Glossary. DSSResources.COM, World Wide Web, http://DSSResources.COM/glossary/

Pyle D (1999) Data Preparation for Data Mining. San Francisco/CA, USA.

Roberts SW (1959) Control chart tests based on geometric moving averages. Technometrics 1: 239–250.

Rook FW & Coghan JW (1989) The Knowledge Acquisition Activity Matrix: A System Engineering Conceptual Framework. IEEE Transactions on Systems, Man, and cybernetics 19(3): 586–597.

Runger GC & Testik MC (2004) Multivariate Extensions to Cumulative Sum Control Charts. Quality and Reliability Engineering International 20: 587–606.

Sachs E, Ingolfsson A, & Ha S (1990) Tuning a process while performing SPC: an approach based on the sequential design of experiments. In: IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop: 126–130.

Sachs E, Hu A, & Ingolfsson A (1995) Run by run process control: combining SPC and feedback control. IEEE Transaction on Semiconductor Manufacturing 8(1): 26–43.

Seabra Lopes L & Camarinha-Matos LM (1995) A Machine Learning Approach to Error Detection and Recovery in Assembly. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol 3: 197–203.

Simpson WR & Sheppard JW (1994) System Test and Diagnosis. Kluwert Academic Publishers.

Spanos CJ (1992) Statistical process control in semiconductor manufacturing. Proceedings of the IEEE, 80(6): 819–830.

Spanos CJ & Chen RL (1997) Using Qualitative Observations for Process Tuning and Control. IEEE Transactions on Semiconductor Manufacturing 10(2): 307–316.

Stout RDL, Van de Goor AJ, & Wolff RE (1998) Automatic fault localization at chip level. Microprocessors and Microsystems 22: 13–22.

Stuart M, Mullins E, & Drew E (1996) Statistical quality control and improvement. European Journal of Operational Research 88: 203–214.

Su LM, Zhang H, Hou CZ, & Pan XQ (2002) Research on an Improved Genetic Algorithm Based Knowledge Acquisition. In: 1st International Conference on Machine Learning and Cybernetics: 455–458.

Sutton JC (1992) Identification of Electronic Component Faults using Neural Networks and Fuzzy Systems. In: International Conference on Industrial Electronics, Control, Instrumentation, and Automation, vol 3: 1466–1471.

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

Suck T (2001) AOI in the spotlight. OnBoard Technology: 40–43.

Tong DW, Walther E, & Zalondek KC (1989) Diagnosing an analog feedback system using model-based reasoning. In: Annual AI Systems in Government Conference: 290–295.

Turban E (1991) Managing Knowledge Acquisition from Multiple Experts. In: IEEE/ACM International Conference on Developing and Managing Expert System Programs. pp. 129–138.

Turban E (1995) Decision support and expert systems: management support systems. Englewood Cliffs, N.J., Prentice Hall.

Ulieru M (1993) From Boolean Probabilistic to Fuzzy Possibilistic Fault Tree Processing in Diagnostic Decision Making. In: European Congress on Intelligent Techniques and Soft Computing, vol 1: 408–414.

Ulieru M (1994) A Fuzzy Logic Based Computer Assisted Fault Diagnosis System. European Journal of Diagnosis and Supervision 3: 405–441.

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

Wagner LC (2001) Failure analysis challenges. In: 8th International Symposium on the Physical and Failure Analysis of Integrated Circuits: 36–41.

Wang HC & Wang HS (2005) A hybrid expert system for equipment failure analysis. Expert Systems with Applications 28: 615–622.

Weiss SM & Indurkhya N (1998) Predictive Data Mining, a practical guide. San Francisco/CA, USA.

West DA, Mangiameli PM, & Chen SK (1999) Control of complex manufacturing processes: a comparison of SPC methods with a radial basis function neural network. Omega International Journal of Management Science 27: 349–362.

Wills CE (1994) User Interface Design for the Engineer. In: Electro/94 International Conference.

Wilson DJH & Irwin GW (1998) Multivariate SPC using radial basis functions. In: International Conference on Control, UKACC'98, vol 1: 479–484.

Winter GB (1992) An Automated Knowledge Acquisition System for Model-Based Diagnostics. In: Autotest Conference Record: 167–174.

Xing D & Li X (1992) Open model methodology: a new approach to the development of user interfaces based on knowledge processing. Software Engineering Journal: 230–236.

Yang Y, Lu N, Wang F, & Ma L (2002) A new fault detection and diagnosis method based on principle component analysis in multivariate continuous processes. In: 4^{th} World Congress on Intelligent Control and Automation: 3156–3160.

Zhang C, Deng H, & Baras JS (2000) Comparison of Run-to-run Control Methods in Semiconductor Manufacturing Processes. Technical Research Report, T.R. 2000-33, Institute for systems Research.

239. Torkko, Margit (2006) Maatilakytkentäisten yritysten toimintamalleja. Laadullinen tutkimus resursseista, kehittymisestä ja ohjaustarpeista

240. Hämäläinen, Matti (2006) Singleband UWB systems. Analysis and measurements of coexistence with selected existing radio systems

241. Virtanen, Jani (2006) Enhancing the compatibility of surgical robots with magnetic resonance imaging

242. Lumijärvi, Jouko (2006) Optimization of critical flow velocity in cantilevered fluid-conveying pipes, with a subsequent non-linear analysis

243. Stoor, Tuomas (2006) Air in pulp and papermaking processes

244. György, Zsuzsanna (2006) Glycoside production by in vitro *Rhodiola rosea* cultures

245. Özer-Kemppainen, Özlem (2006) Alternative housing environments for the elderly in the information society. The Finnish experience

246. Laurinen, Perttu (2006) A top-down approach for creating and implementing data mining solutions

247. Jortama, Timo (2006) A self-assessment based method for post-completion audits in paper production line investment projects

248. Remes, Janne (2006) The development of laser chemical vapor deposition and focused ion beam methods for prototype integrated circuit modification

249. Kinnunen, Matti (2006) Comparison of optical coherence tomography, the pulsed photoacoustic technique, and the time-of-flight technique in glucose measurements *in vitro*

250. Iskanius, Päivi (2006) An agile supply chain for a project-oriented steel product network

251. Rantanen, Rami (2006) Modelling and control of cooking degree in conventional and modified continuous pulping processes

252. Koskiaho, Jari (2006) Retention performance and hydraulic design of constructed wetlands treating runoff waters from arable land

253. Koskinen, Miika (2006) Automatic assessment of functional suppression of the central nervous system due to propofol anesthetic infusion. From EEG phenomena to a quantitative index

254. Heino, Jyrki (2006) Harjavallan Suurteollisuuspuisto teollisen ekosysteemin esimerkkinä kehitettäessä hiiliteräksen ympäristömyönteisyyttä

Book orders:
OULU UNIVERSITY PRESS
P.O. Box 8200, FI-90014
University of Oulu, Finland

Distributed by
OULU UNIVERSITY LIBRARY
P.O. Box 7500, FI-90014
University of Oulu, Finland

UNIVERSITY OF OULU P.O. Box 7500 FI-90014 UNIVERSITY OF OULU FINLAND

ACTA UNIVERSITATIS OULUENSIS

ISBN 951-42-8204-3 (Paperback)
ISBN 951-42-8205-1 (PDF)
ISSN 0355-3213 (Print)
ISSN 1796-2226 (Online)