# Test Result Archiving and Diagnosis systems

Praveen Sirikumara

Master's Thesis in Computing Science, 20 credits
Supervisor at CS-UmU: Michael Minock
Examiner: Per Lindström

UMEÅ UNIVERSITY
DEPARTMENT OF COMPUTING SCIENCE
SE-901 87 UMEÅ
SWEDEN

**Abstract**

An expert system is expected to imitate the expertise of a human expert. A diagnosis system is a special type of expert system which, is used to establish the cause of a malfunction in a complex situation based on observed symptoms. The knowledge representation plays a key roll in such knowledge based systems' performance like human experts. Therefore, the concepts like rule-based reasoning, case-based reasoning, model-based reasoning, knowledge engineering, knowledge representation schemes are discussed in this thesis. Further, the aspects of designing a diagnosis system are also discussed.

# Contents

# List of Figures

# Chapter 1

# Introduction

Quality assurance plays a key roll in product manufacturing processes. A product goes through a number of stages from order to delivery during these processes. Quality should be assured during all stages. This assurance starts at establishing the requirements of the product and documenting them, and then manufacturing is carried out according to the document of requirements. Furthermore all the material and components involved in the manufacturing processes are documented and controlled. It is also essential that the equipments and procedures utilized during these processes are calibrated and correctly maintained. Another aspect is that the staff members involved in all stages has to be well trained and qualified. Additionally, the procedures carried out during these processes have to be documented and archived for later retrieval. When it comes to delivering the product, it is expected to fulfill the requirements made at the beginning of the processes. Therefore a higher level of quality assurance, a good system of documentation of the manufacturing processes is essential.

A database management system can be used to store the information and data that should be included in quality assurance documents. A product is tested for different functionalities before it is completely manufactured. This helps to locate faults in a product in the early stages and resolve them. In most of the cases, causes for such faults are known and they can be resolved easily given the necessary instructions are available. A diagnostic system with a knowledge base about the known faults encountered with products during manufacturing and the causes for such faults is an asset in the manufacturing processes.

This thesis presents in-depth studies of diagnostic system which is a special type of expert systems, reasoning mechanisms used in expert systems, knowledge engineering and knowledge representation schemes. Furthermore a description is presented about the processes of implementing a prototype of an archiving database and a diagnostic system for Amersham Biosciences AB. The diagnostic system in this case is provided with the symptoms (i.e. bad test results) and information of a faulty product or manufacturing process and suggests the most suitable way to fix the faulty product.

## 1.1 The Company

Amersham Biosciences AB is a company developing technologies and products for genomics, proteomics, bio-arrays, protein separations and informatics. The company is a part of GE Healthcare. GE Healthcare is a General Electric Company that is headquartered in the United Kingdom. Worldwide, GE Healthcare employs more than 42,500 people committed to serving health care professionals and their patients in more than 100 countries. There are a number of bio-technical and medical equipments and instruments manufactured at Amersham Biosciences Umeå plant. Amersham Biosciences Umeå plant has about 420 employees and has been constructing and manufacturing advanced bio-technical and medical products for thirty-five years. Quality assurance is a highly observed property in their manufacturing process. In addition, a great deal of effort is made to provide a higher level of quality assurance to their customers.

Consequently, the products are tested for accurate functionality during and after the manufacturing processes. These tests are carried out on the components of the product as well as on the completed product. Most of the tests are automated or semi-automated and these tests are carried out with the help of computer programs. Most of these programs are written in the programming environment Agilent VEE Pro.

Presently, when a product is tested the program that carries out the test saves the results in files. Therefore the data from the test results of the products are managed by programming with files, i.e. file processing. Aiming for higher level of quality assurance, this procedure of archiving is to be developed into the next phase where the test programs written in Agilent VEE Pro can store the results in a database. Oracle has been chosen by the company as the preferred database management system.

## 1.2 Benefits of Knowledge Systems

Sometimes a product that is being tested for functionality would fail a test. The cause for this failure is often known to the engineer who designs the test but not to the technician who carries out the test. The cause could be because of some fault in the product or some problem in the manner the test was carried out. The possibility of using the engineer's expertise when a technician encounters a failure with a test would be highly desirable. Therefore an appropriate diagnostic system is of interest for a better manufacturing process.

A knowledge system can be used to capture the expertise which different individuals possess. This expertise can be distributed to be used where the human expert is not available.

# Chapter 2

# Background

## 2.1 Expert Systems

Expert systems are used to imitate a human expert's methodology and performance in solving a problem in a certain domain. An expert system can be divided into several components; the user interface, knowledge base, inference engine, explanation subsystem and knowledge base editor.

**User interface** The communication between the user and the system is simplified by providing a user interface. Furthermore, the complex structures in the system implementation are hidden to the user. The interface facilitates the user with communication though question-and-answer, menu-driven or graphical interface styles.

**Knowledge base** Knowledge base is regarded as the heart of the expert system. A good knowledge base is essential for a good expert system. The knowledge of the particular domain is represented in this base. It holds both general knowledge and case-specific knowledge. The information about the current problem instance like data given by the user, partial conclusions, confidence measures and dead ends in the search process are regarded as case-specific data.

**Inference engine** The application of the knowledge in the knowledge-base to the problem instance is carried out by this component of the system.

**Explanation subsystem** This component is used to explain the reasoning process during problem solving to the user. Questions like why a certain conclusion is made how the conclusion was made can be described with help of an explanation subsystem.

**Knowledge-base editor** The Perception of human experts subject to develop and change during a time period. Further, the initial knowledge in the knowledge-base often needs to be revised. In this both cases the editor is used to change the knowledge stored in the knowledge base.

This modulation provides good foundation and interface to improve an expert system. The separation of knowledge base and the interference engine is desirable because of following reasons [4].

– Knowledge can be representation in a more natural way, rather than with computer codes which would be difficult to comprehend.

– The knowledge can be stored without considering the aspects of low-level details of computer implementation.

– A change in knowledge base would not affect the inference engine controls or vice-verse.

– The same inference engine can be used in different systems that are intended to different domains.

Expert systems are developed in variety of domains like medicine, mathematics, engineering, chemistry, geology, computer science, business, law, defense and education. These systems deal with different types of problem categories, which can be classified as [4]:

**Interpretation** This category of expert systems makes high-level conclusions from collections of raw data

**Predication** The probable consequences of given situation is projected in these expert systems.

**Diagnosis** These systems are capable of establishing the cause of a malfunction in a complex situation based on observed symptoms.

**Design** This category deals with finding a configuration of system components that meets performance goals while satisfying a set of design constraints.

**Planning** When provided start conditions and run-time restrictions a sequence of actions are planed by this systems to achieve a set of goals.

**Monitoring** These systems are capable of comparing the behavior of a system with its expected behavior.

**Instruction** The characteristic of these systems is the capability of assisting during education process in technical domains.

**Control** This category of systems is able to direct the behavior of a complex environment.

## 2.2 Reasoning Mechanisms

The reasoning mechanism embedded in the expert system is a central concept. Therefore, the following three sections are dedicated to present introductions to rule-based reasoning, case-based reasoning and model-based reasoning.

### 2.2.1 Rule-based Reasoning

Rule-based reasoning is a widely used and most natural reasoning mechanism [4]. It falls into the deductive reasoning category. The characteristic property in this reasoning mechanism is that it uses established domain knowledge as ”if ... then ... ” rules. This is an effective methodology when the fundamental theories of the domain are recognized [2]. The rule-based reasoning can be conducted as goal-driven or data-driven.

The goal driven method is to start with a hypothesis and examine whether the preconditions are satisfied to confirm the hypothesis. That is, reasoning is started by assuming the ”then” part of an ”if ... then ... ” rule is to be true. Then the mechanism checks whether the ”if” part of that rule can be established. To do that, ”then” parts of a new rule can be assumed to be true. These assumptions and partial proofs are recorded in the case-specific data in the knowledge base. The user is expected to provide information to prove a certain rule. When a certain assumption is proved the rule that has that fact as the precondition can be established. This leads to the primer assumption to be proved, which makes it the conclusion of the situation.

The data-driven method starts with the precondition and establishes the conclusion. In other words the observations of the situation are compared with the ”if” parts of the rules and the ”then” parts are established. These new preconditions are then tested against the rules to derive new information. These are several ways to decide how the rules are tested against the case-specific data derived by the reasoning mechanism. It can be the first rule that has preconditions satisfied or a random one, it can be the most specific rule (the rule having the most preconditions satisfied). Another way is assigning a weight for the rules for their compatibility and choosing the rule with highest weight from the rules that are satisfied. These rules can be searched using either depth-first or breadth-first search.

### 2.2.2 Case-based Reasoning

In the case where the domain knowledge is not that established, case-based reasoning can be a good candidate. This method is based on the knowledge of previous cases to draw conclusions about new cases. Case-base reasoning is an inductive reasoning method [2]. The known cases are stored in an explicit database and the new cases can be added to the database. Thereby this method learns with time. When it comes to solving a new problem the case-base reasoning method use the following steps [4];

1. **Retrieving suitable case from the database** Finding a similar case from the past situation is a difficult task because of the difficulty of defining the similarity

between two cases. Organizing the storage of the cases and retrieval of cases is central for effective case-based reasoning method. There are several heuristics to cope with this. Cases can be organized by the goal and retrieved when the case has the same goal as the current situation. Another organizing method is to use cases with most important features matched or the most number of features matched. The matching may first look for exactly matched case before looking for a more general case. Using the cases most frequently matched or most recently matched is also used when retrieving cases to match a new situation. Another method is to use the case that matches without much adjusting. Using these heuristics a similar case is retrieved.

2. **Adjust a retrieved case to the current situation** An already established case contains the operations for transforming the initial state to the goal state. The reasoning mechanism has to determine the operations to transform the established case to the current situation. If analytical methods are available then they are used otherwise heuristic methods are applied to find out the operations for the new case.

3. **Utilizing the adjusted case** The modified case is tested against the current situation and adjusted until it successfully solves the current situation.

4. **Saving the solution with information about the success** This requires updating the indexing structure. This is beneficial for solving similar problems in the future. Further, the system is learning by adding new cases this way.

### 2.2.3 Model-based Reasoning

The characteristic of model-based reasoning is that the functionality and specification of the system in question are used to draw conclusions about it. Model-based techniques tend to represent knowledge more completely but briefly and at a greater level of detail than techniques that encode experience. This is because they employ models that are compact axiomatic systems from which large amounts of information can be deduced. The model-based reasoning often simulates the functionality of the situation it should be able to cope with. This mechanism is often used in the educational systems and diagnostic systems. These systems are intended to deal which physical devices that have predefined behaviors. If the behavior differs from the prediction a model-based reasoning system with following properties can identify the fault [4].

- A description of the components in the system which would provide the ability to simulate the behavior of each component.

- An account of the interconnection between the components of the system. This leads to the possibility of simulating the interactions in the system

- The observations and measurements made, when the system was functioning properly.

  - The rules connecting the observation and the faulty component which cause the observation

The model-based reasoning approach starts with assuming some components in the system are defected based on the observations. Then those components are tested for functionally and the faulty components or the component are located.

## 2.3 Knowledge Engineering

Knowledge engineering is the subfield of Artificial intelligence concerned with applying knowledge to solve problems that ordinarily require human intelligence. Knowledge engineering has the following steps [8]:

**Identify the task:** The range of questions that the knowledge base should be able to answer must be established. Further, the facts that should be included in the knowledge base have to be identified.

**Assemble the relevant knowledge:** The knowledge that should be included in the knowledge base is identified. This identification is done with regards to the task. Consequently, all the knowledge in the domain does not need to be included to the knowledge base if that is not required by the task.

**Decide on a vocabulary:** A vocabulary is chosen to represent the concepts of the domain. This vocabulary is known as the ontology of the domain. The ontology describes the existence of things but not their properties or interrelationships.

**Encode general knowledge about the domain:** General knowledge in the domain is included to the knowledge base. A good ontology will result in less general information in the knowledge base.

**Encode a description of the specific problem instance:** Information about specific situation is added to the knowledge base.

**Pose queries to the inference procedure and get answers:** New knowledge about a specific problem is obtained with the help of inference procedures and the problem specific facts in the knowledge base.

**Teach the knowledge base:** The incorrect knowledge derived from the knowledge base has to be corrected by completing the knowledge base with general information and problem specific facts.

### 2.3.1 Knowledge Representation

This concept could be simply introduced as formalizing and organizing the knowledge. The concept is best describes in five points [7]. That is a knowledge representation is:

a replacement: The reasoning process utilizes the knowledge representation to reason about entities in the world. The entities that exist in the world have to be denoted in the reasoning process. As a result the knowledge representation acts as a replacement or, a surrogate, for the entities in the world of interest.

a set of ontological commitments: Entities in the represented world may have many properties and behaviors. But a representation does not hold every property and behavior about the entity; thereby a choice is made about how to see the world. That is, making a set of ontological commitments.

an incomplete theory of intelligent reasoning: A representation is usually based on some belief about how reasoning is carried out intelligently. This is incomplete because intelligent reasoning is a compound process which is difficult to comprehend and the representation of this belief also ignores some insignificant aspects of the belief.

a medium for efficient computation: Reasoning is considered to be a computational process carried out on the representation. Therefore a representation is made to be efficient in the computational aspect.

a medium of human expression: When it comes to expressing the world to a machine, by a human, the knowledge representation is the medium that is been used to teach, or tell, the machine about the world.

### 2.3.2 Knowledge Representation Schemes

Knowledge representation schemes can be defined as an adequately precise notation for representing knowledge. These schemes can be categorized as *logical representation schemes*, *network representation schemes* and *procedural representation schemes*. Logical and network representations schemes have the common property of being declarative and therefore can be considered as declarative representation schemes. Most of the implementations of knowledge schemes tend to have some combination of these three categories. [6]

#### Logical Representation Schemes

Main notions occupied in these schemes are constants, variables, functions, predicates, logical connectives and quantifiers. When it comes to knowledge base, logical formulas are the atomic unit. In other words, a collection of logical formulas which try to illustrate the world constitutes as the knowledge base. The knowledge base can be modified through introduction and deletion of logical formulas.

There are advantages that seem to be direct consequence of the logic formulas, which these schemes are based on. In the case of information retrieving, semantic constrain checking and problem solving procedures the proofs can be based on the inference rules that the logical terms constitutes. Another advantage is the formal semantic of logic, which is considered to being clean, well-understood, and well-accepted. This is the case when the representation is close to First Order Logic. The descriptions in the knowledge base become understandable because they are based on simple notions. In addition, the facts are represented once, independent of the different use of them.

When it comes to disadvantages, it is difficult to organize the facts in the knowledge base because such principles are lacking. Additionally, representing the procedural and heuristic knowledge seems to be complicated.

### Network Representation Schemes

These schemes are also referred to as semantic networks. The characteristic of this representation scheme is that world is represented with objects and associations. The objects represent individuals and binary associations of binary relationships in the world. In this representation the knowledge base is a collection of objects and associations.

These schemes have different association types, namely classification, aggregation, generalization and contexts. But it is not necessary that one scheme contain all these types.

**Classification:** Reflects that an object has a generic object and therefore the object is an instance of the generic object.

**Aggregation:** Refers to an object and its components. This way an object is a part of an object.

**Generalization:** An object is associated to more generic object.

**Contexts:** The scheme is partitioned and objects and associations hold a hierarchical position in the network.

When it comes to traversing the knowledge base the associations are used to define paths. This is an advantage in the information retrieval process. Furthermore, this kind of knowledge base can be organized easily according to above mentioned association types. Another advantage of network representation schemes is that the knowledge base can be represented graphically and this often leads to better understanding of knowledge base.

As for the drawbacks of these schemes, lack of formal semantics and standard terminology can be mentioned.

### Procedural Representation Schemes

The main characteristic of these schemes is that the knowledge base consists of a collection of procedures which describe how to solve a problem instance.

Due to direct interaction with facts the need of searching is eliminated in procedural representation. This is regarded as a major advantage of this representation. On the other hand, the difficulty of understanding and modifying the knowledge base can be mentioned as for a drawback.

### Frame-based Representation Schemes

Frames are considered to be a compound data structure which represents a situation. They contain slots to represent objects that comprise the situation and the relationships between the objects in the situation. Frame-based schemes are based on ideas combined from declarative and procedural representations. The knowledge base in this representation is a collection of frames which are organized according to some combination of association types described under network representation schemes.

## 2.4 Graphical Models

A major problem in knowledge-based systems is maintaining the knowledge in an effective manner. The properties of the entities and the relationships between them in the knowledge base should be inserted, updated and deleted. These procedures are not considered difficult when the knowledge representation scheme support organizing techniques for the knowledge base. Network representation schemes discussed in Section 2.3.2 can be considered as a graphical representation of the knowledge. Graphical models are another suitable candidate for maintaining knowledge in a knowledge system.

A knowledge system contains complex structures of knowledge about the domain it represents. When obtaining an output from the knowledge system depending on the initial inputs provided, inputs may play different roles on the outcome. Therefore the representation of the knowledge with the dependencies among the entities of the knowledge base is highly appreciated. Graphical models provide the properties of both graph theory and probability theory. Thereby graphical models are capable of maintaining the complexity and uncertainty of knowledge of a certain domain. Complexity can be simplified with the modular approach that the models provide. A complex system is build by combining simpler modules. The elements of the probability theory in the model cover the uncertain relationships between these modules. The graph theory properties of the graphical models provide humans with an intuitive interface to interact with and represent the entities and their relationships in the domain's knowledge.

There are three types of graphical models; [5]

**Directed graphical models** Bayesian networks, belief networks, generative models, causal models are directed graphs.

**Undirected graphical models** Markov networks and Markov random fields are undirected graphs.

**Chain graphs** Both directed and undirected arcs can be used in these graphs.

The ability to represent the causal relationship between entities makes directed graphical models a perfect candidate for representing knowledge. In the following section Bayesian network which is a directed graph is presented.

### 2.4.1 Bayesian Networks

A Bayesian network can be defined as follows; [8]

- Nodes of the network are constituted by a set of random discrete or continues variables.
- Nodes are connected by arrows. An arrow from node *X* to node *Y* implies that the *X* is the parent of *Y*.
- Each node has a conditional probability distribution which constitutes the effect by its parent nodes.
- Bayesian networks are acyclic, directed graphs also known as DAG.

Relevant knowledge of a domain can be represented as a Bayesian network. The knowledge engineer would firstly, arrange the nodes and the arrows to represent the dependencies between nodes, which represent different variables, in the domain. Thereafter, the conditional probability distribution for each variable, with regards to its parents is established on respective arrow. This arrangement of nodes and arrows of the network and the conditional probability distribution is made while the full joint distribution for all variables is preserved.

## 2.5 Diagnostic Systems

It is often difficult to establish a model which illustrates a situation which would rise in a certain domain. But luckily, in some cases a model or scientific theorems which describe a domain is already established. In such domains, the behavior of a system with certain component can be predicted. If a system, built on an established model, differs from the expected behavior the reason for this could be determined with the help of the model. A system that uses this model to discover the reason for the anomaly in the system is known as a diagnostic system. A diagnosis system is one that is capable of identifying the nature of a problem by examining the observed symptoms [1]. Mechanics, electrics and electronics are examples for domains that have established models.

### 2.5.1 Diagnostic Of Mechanical/Electrical Faults

Since the systems from mechanics, electrics and electronics are designed and constructed by humans the fundamental knowledge about the system is available. Thereby an irregularity in the system can be tracked back to the faulty component in the system. Human experts know the relationships between components in the system and which consequences would be apparent due to a malfunction in a component. If these relationships can be represented in a knowledge base, a diagnostic system can be used to make diagnosis of faulty systems.

### Knowledge Acquisition

Experts in domains like electronics possess specific and high-quality factual and heuristic knowledge in diagnosing faulty systems. This knowledge has to be gathered in order to create a knowledge base. This process, also known as knowledge acquisition, is considered to be a major problem in knowledge engineering. When it comes to diagnostic systems, knowledge acquisition methods can be divided into three major techniques: deductive, inductive and experience-based techniques. [9].

**Deductive techniques:** The common property of these techniques is to start with a symptom that a faulty system could exhibit. Then the possible causes or faults are identified. These relationships are represented as a tree where the root is the failure and the leave nodes are the primary causes. Often these trees have several levels of nodes, which represent secondary causes and their symptoms. These techniques are also known as *fault tree analysis*.

**Inductive techniques:** These techniques begin with possible failures in a system. Thereafter, the symptoms that would arise from each failure are recognized. Normally systems are divided into sub systems and how a problem in a sub system would reflect in the system is recognized. The techniques are also known as *failure and effect analysis*.

**Experience-based techniques:** The interest in these techniques lies on investigating the past failures and the cause of them. Often several causes are examined and the relationship between the causes and the main failure is recognized.

## 2.6 Human Fault Diagnosis model

There are several models that are used to isolate a failure in a system by a human. Those are a process model, a resource model, a failure model and an observable model [9]. A process model illustrates functional aspects of the system. Process steps, process transitions and process faults are described in the process model. The process steps can be organized according to the order they occur in a system. Thereby the process transitions can be organized which provide the possibility to organize the process faults. The process faults comprise the process specific failure cause. The resource model corresponds to the physical structure of a system. In this model the components in a system are ordered based on the component type. These component types are related to component type specific failures. The failure model represents the failure cause and test procedure to pinpoint such failures. The observation model contains the error codes, internal system address, measurable values and qualitative observations which a system would provide. A human expert starts the diagnosis on a system by looking for error codes or symptoms which are described in the observation model. Such indications often lead to assumptions in which process or component type do the cause lies. This finding is connected with the process step, process transition or component in the system. Therefore tests from the fault model are carried out to identify the faulty process or the component. When the identifications are made the relating faulty cause is presented as the diagnosis. Such diagnosis is based on both theoretical and practical understanding of the system in questions. Further the expert develops tricks, shortcuts and heuristics for making a diagnosis.

# Chapter 3

# Approach

The implementation tasks with regards to my thesis were two folded.

1. **Archiving System** This system is intended to archive data about the products manufactured at Amersham Biosciences AB. The system holds, the results and criteria of the tests carried out on products to guaranty correct functionality, the components that a products consists of, the tools used during the tests, the batch information about products, information about customers, and information about suppliers. This information in the archiving system is to be mainly used to provide quality assurance documents with products and as a data source for business intelligent tasks to improve manufacturing processes.

2. **Diagnosis System** This system is intended to help the technicians who carries out the tests on the products. Occasionally a product could fail on a test carried out to check the functionality. In that case a technician can find out the cause for the malfunction with a good knowledge about the product. But this required knowledge is often possessed by the engineers who designed the product. The diagnosis system is expected to imitate such an expert. The system is to be equipped with the knowledge of faults cause and it is expected to make diagnosis with the answers that the technician provides about the product and the symptoms it shows.

The approaches taken to implement prototypes of these two systems are described in the following sections of this chapter.

## 3.1 Archiving System

The approach taken in designing and implementing the product details archiving system can be divided into six phases as follows;

**Familiarizing with Agilent VEE:** Agilent Visual Engineering Environment (VEE) Pro is an easy-to-use graphical programming environment that helps speedup development of measurement and control, analysis, test-system and data-acquisition applications, used in industries including aerospace, defense, wireless and consumer electronics [3]. The online tutorials and manuals were utilized to get acquainted with the programming environment and to understand the underlying concepts of the programming language.

**Familiarizing with Oracle:** Online tutorials as well as manuals were used to get familiarized with the database management system. These manuals were also helpful during the process of installing and managing the system. Oracle Distributed Object Technology was specially studied to communicate with Oracle through Agilent VEE.

**Market survey:** The goal was to find different user expectations of the system numerous interviews were carried out with different staff members involved in the manufacturing processes; namely control engineers, quality engineer, customer care officer, quality manager and production manager. The key requirements mentioned in these interviews were security, integrity of data and the possibility to generate different types of reports.

**Analyzing data:** Designing a database requires identifying the data structures and underlying relationships. Therefore the data in the files were imported to a temporary database to be analyzed. A script written in the Perl scripting language was utilized to import data from 7636 data files into the database. Structured Query Language (SQL) was utilized to retrieve data from this temporary database to analyze.

**Designing and implementing database:** After the data structures and relationships were identified a schema was made to represent them (see figure 3.1). Thereafter the implementation of the database was carried out preserving the established relationships and structures.

**Creating a user interface:** Agilent VEE Pro was used to create the user interface to the database. This was due to the fact that it was requested to find out, whether it is possible to access database using Agilent VEE Pro, and if so, how suitable it is to create the entire graphical interface to the database using this language. Furthermore; most of the potential users are familiar with Agilent VEE Pro as the test programs for products are written in Agilent VEE Pro.

### 3.1.1 System Description

The conceptual schema of the database created to archive the product details and test results data is shown in Figure 3.1 as an entity relationship diagram, i.e. ER-diagram. Based on this conceptual model the database was designed using 12 tables. The schemas for these 12 tables are shown in Figure 3.2. A description of each table is given. The relationships mentioned in the descriptions are the relationships represented in the figure 3.1 between different entities.

Figure 3.1: Product Details Archive Database - Entity Relationship Diagram

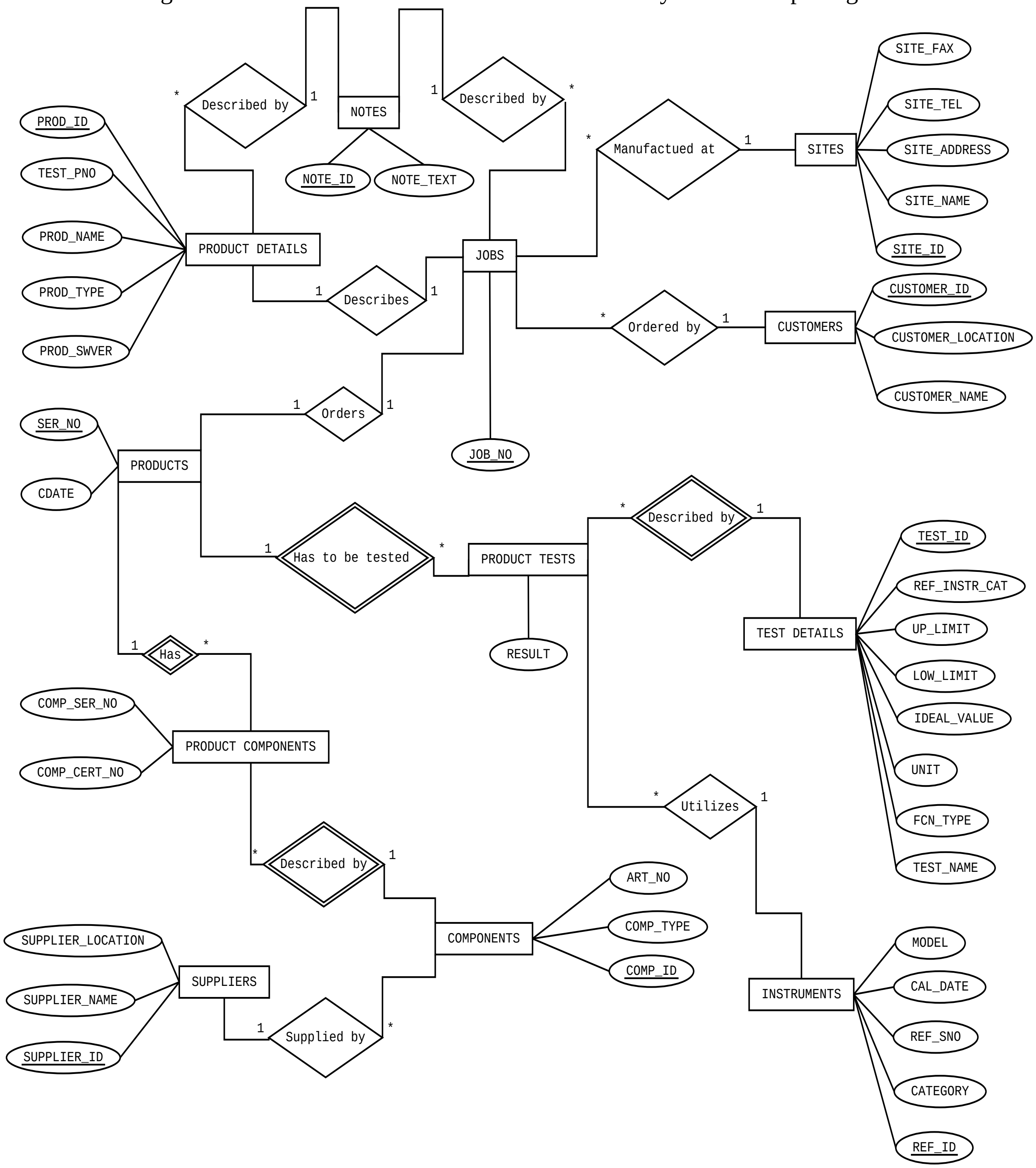


Figure 3.2: Product Details Archive Database - Schema

**Components**

| | | | |
|---|---|---|---|
| Component_id | Supplier_id | Comp_type | Art_no |

**Customers**

| | | |
|---|---|---|
| Customer_id | Cus_name | Cus_location |

**Instruments**

| | | | | |
|---|---|---|---|---|
| Instrument_id | Inst_serial_no | Inst_modle | Inst_category | Calibration_date |

**Jobs**

| | | | | |
|---|---|---|---|---|
| Job_no | Customer_id | Site_id | Note_id | Product_id |

**Notes**

| | |
|---|---|
| Note_id | Note_text |

**Product Components**

| | | | |
|---|---|---|---|
| Prod_serial_no | Com_id | Com_serial_no | Com_Certificate_no |

**Products**

| | | |
|---|---|---|
| Serial_no | Job_no | Date |

**Product Tests**

| | | | | |
|---|---|---|---|---|
| Serial_no | Test_id | Result | Engineer_id | Instrument_id |

**Product Detail**

| | | | | | |
|---|---|---|---|---|---|
| Product_id | Prod_name | Prod_type | Note_id | Prod_software | Test_no |

**Test Detail**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Test_id | Test_name | Function_type | Unit | Ideal_value | Lower_limit | Higher_limit | Instrument_category | Engineer_id |

**Sites**

| | | | | |
|---|---|---|---|---|
| Site_id | Site_name | Site_Address | Site_telephone | Site_fax |

**Suppliers**

| | | |
|---|---|---|
| Supplier_id | Supp_name | Supp_location |

**Components:** Information about the component which would be used in different products is stored in this table. The supplier of the component is identified by referring to the supplier in the supplier table and the relationship *supplied by* between the supplier and components are preserved. The component type and article number is also recorded in this table.

**Customers:** Customer name and customer location is recorded here. Customers are referred from the jobs as they place orders for different products.

**Instruments:** Information about the instruments used to test the functionality and accuracy of the products are stored in this table. The serial number of the instrument, instrument category, model details and calibration date are among the information recorded about an instrument. Furthermore, with every new calibration of the instrument it is given a new id, which the test programs using the instrument would refer to in the test result record in the product tests table.

**Jobs:** This table holds the information about the orders placed by the customers. Each order has a unique job number and refers to a customer record from the customer table. Thereby preserves the relationship *ordered by* between jobs and the customer. The site, where the manufacturing is carried out, is referred to in the site table to represent the relationship *manufactured at*. If there is any specific information that should be made available for the customer, this information is referred from the notes table. Thus *described by* relationship between jobs and notes is maintained. Furthermore the product that was ordered by the customer is referred from the product detail table. Consequently, the relationship *describes* is represented.

**Notes:** Information or notes about a job or a product is stored in this table. These notes are often a description or a statement that should be included with the quality assurance documentation provided with the product.

**Product components:** Details about the components of products are recorded here. Each product is identified with the product serial number and the components are recorded with their serial number and certificate number. The components description is identified by storing the components' id:s in the table. This establishes the *described by* relationship between components and product components in the database.

**Product:** The product serial number and date of issue for the quality assurance documentation are recorded in this table. Each product has to have a customer order. This relationship is supported by referring to the job record in the jobs table. This is the representation of the relationship *orders* between jobs and products in the database. By having the product serial number here the *has* relationship is preserved in the database.

**Product tests:** The tests performed on the products and their test results are recorded in this table. The serial number of the product, on which the test is carried out, is stored in the table. This maintains the *has to be tested* relationship between the products and product tests in the database. Information about the tests is referred to by storing the test id in the table, thereby constituting the *describes by* relationship between product tests and test details. Results of the test are numerically represented. Identity of the engineer who performed the test is also recorded. Furthermore, in the case of an instrument is used to test a product the instrument is referred from the instrument tables which represent the *utilizes* relationship.

**Products details:** Information like product name, product type and information about the software included in the product is recorded here. Notes or information about the product is referred to notes table. This is to constitute the *described by* relationship between product details and notes. Records of job table refer to the product details that are included in the customer order, by referring to records of this table.

**Tests details:** Information about the tests preformed on the products is recorded in the table. These records contain test name, test type, unit of the test result, ideal test result value, lower limit of the result value, upper limit of the result value, category of the instrument, and the identity of the engineer.

**Sites:** Information about the sites where products are manufactured is recorded here. The manufacturing site's name, address, telephone number, fax number are recorded here.

**Suppliers:** In this table information like supplier's name and location are stored in this table. These are the suppliers who supply the components.

## 3.2 Diagnostic System

A test carried out to check a certain functionality of a product can fail to succeed. Then the technician who carries out the test can check for minor failures on the product which would cause the test to fail. A technician can make an accurate diagnose if the knowledge about the test and possible causes for the test to fail is available. But often this knowledge is possessed by the engineer who designed the test. If this expertise was available to the diagnostic process it would be more effective.

The approach taken in designing and implementing the diagnosis system can be divided into five phases as follows;

**Understanding the testing process:** Firstly, the process of testing the product for correct functionality was examined. Different products have mostly different test processes. There are several test processes that are common to all of the products. Even with these common tests a particular symptom could have different causes.

**Knowledge acquisition:** During this phase an expert in test process designing was interviewed. Three test processes were chosen and with the experts help these test processes were examined in details. The different symptoms and possible causes for failure in particular test process were identified and documented.

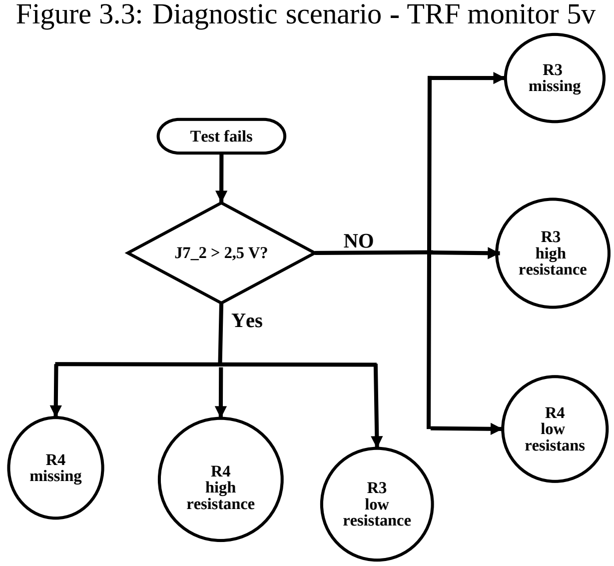
**Encoding knowledge:** The knowledge acquired for each test process was encoded in to flowcharts before translation them in to respective decisions tree.

**Design and Implementation:** Encoded knowledge was inserted into the knowledge base and the diagnosis system was designed and implemented as a combination of case-based reasoning and rule-based reasoning. The knowledge about each test process for respective product was encoded in to the knowledge base with support to case-base reasoning mechanism. The knowledge about a particular test process was encoded with support to rule-base reasoning mechanism.

**Creating a user interface:** Finally, a user interface was created, where a user could provide the product and the test process in the questions to get diagnosis help to find out the cause to the failure.

The Figures 3.3, 3.4 and 3.5 represents three simple diagnostic scenarios developed from the three test process chosen during the knowledge acquisition phase. The flow of the diagnostic process is represented with the help of flowcharts.

Figure 3.3: Diagnostic scenario - TRF monitor 5v




The diagnostic process can also be described by using decision trees for further clarification. The Figure 3.6 is a representation of the flow chart shown in the Figure 3.4 as a decision tree. Such decision trees are recorded into a database as the knowledge base. The knowledge base is organized according to the model of products and the test process that the products are tested for. Therefore the user is expected to provide the system

Figure 3.4: Diagnostic scenario - Flash gate A high C2 low

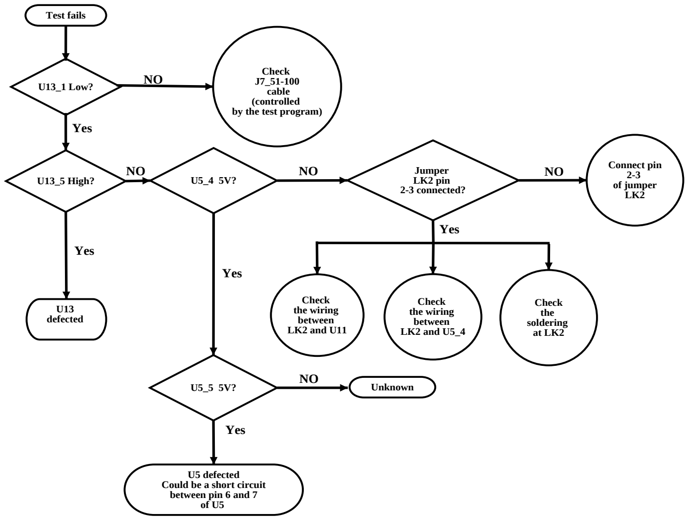



Figure 3.5: Diagnostic scenario - Shutter 1 Active 5v to J2, C5 low C0 high

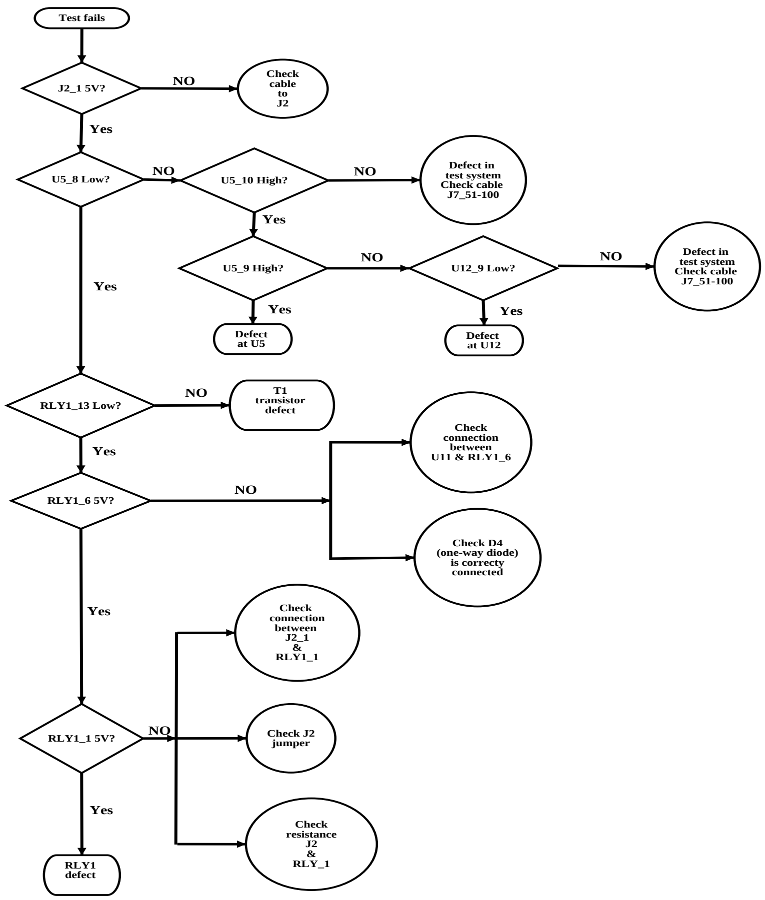
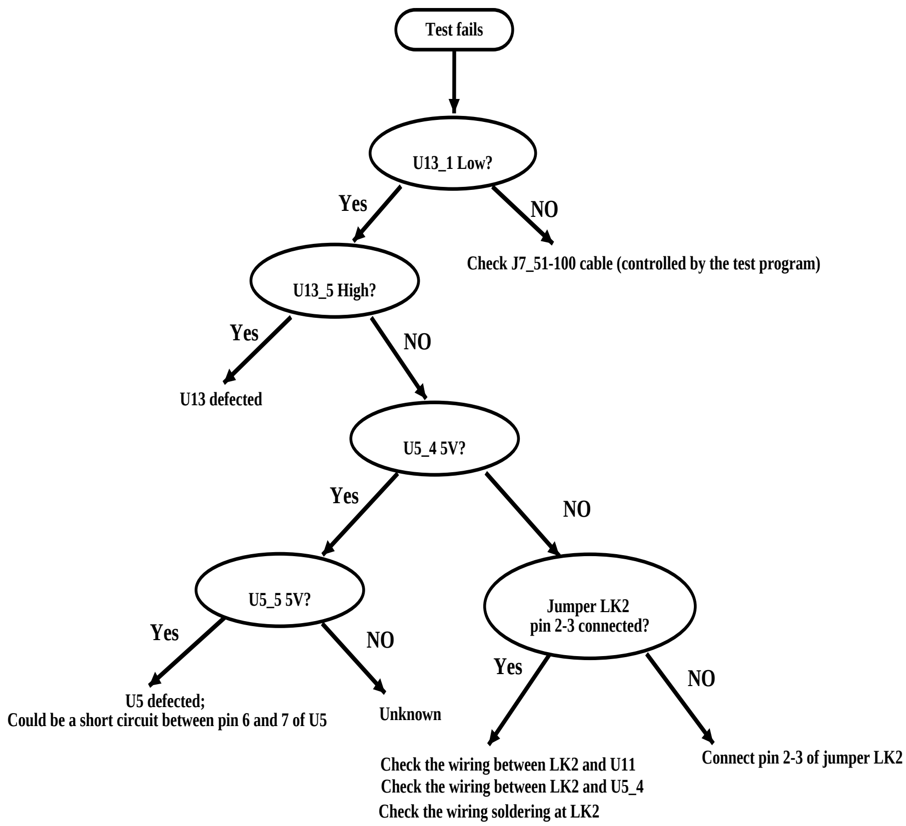


Figure 3.6: A Decision Tree - Flash gate A high C2 low

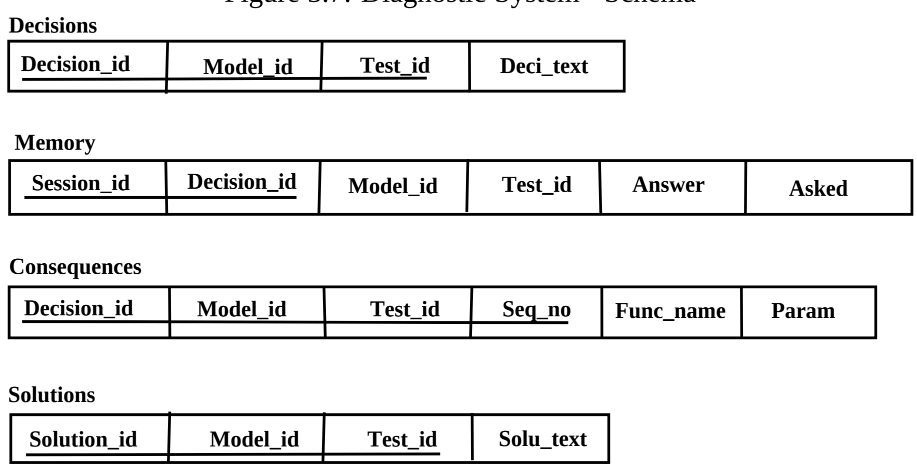


with information about the product model and the test process name to start the diagnosis. When this information is provided the system would ask about the functionality of components of the product. These questions are based on the underlying decision tree of the situation. Based on the answer given by the user the diagnosis would be made. For example the diagnostic scenario described in the Figure 3.4 the questions would be asked as in the same manner as shown in the Figure 3.6. That is, the answer to the question at the parent node in the decision tree, determines the question to be asked at the child node. When an answer to a question leads to a leaf node on the decision tree that information is the given as the cause for the failure of the test process. Now the user can resolve the problem with the product that caused it to fail the testing process. A web based prototype of a diagnostic system was implemented test this approach. Web programming was carried out with PHP and the underlying database management system is MySQL.

### 3.2.1 System Description

A prototype of a diagnostic system was created based on the knowledge represented in the flowcharts. A database was used to store knowledge about each and every known problem and the working memory of the diagnostic process. The inference engine was implemented with help of PHP. In this case the working memory would be the history of the responses given by the user in a certain case of diagnostic. The database is fundamentally based on the schema shown in Figure 3.7.

Figure 3.7: Diagnostic System - Schema




**Decisions:** The questions that would pinpoint the causes of problem are stored here. This information is based on the product model and test identity.

**Consequences:** The information on the consequences of an answer to a question is recorded here. The consequences are first marking the asked question as asked in the working memory to prevent it from repeating. Thereafter a new question is added to the working memory based on the answer of the current question.

**Memory:** Hold the information about the answer to an asked question and the questions to be asked to find the cause in a certain diagnostic session. Further the information about the established solutions is also stored here.

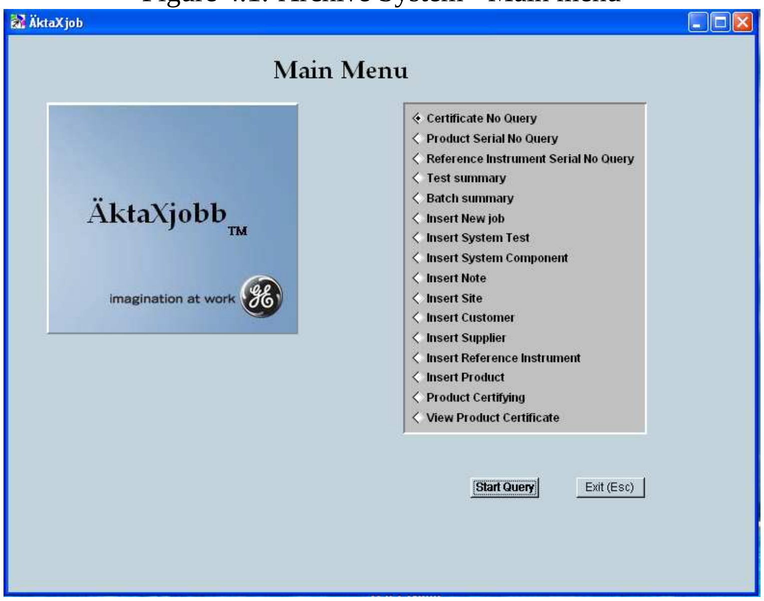
**Solutions:** The texts describing the causes of problems are stored here based on the model and the test identity.

# Chapter 4

# Results

## 4.1 Archiving System

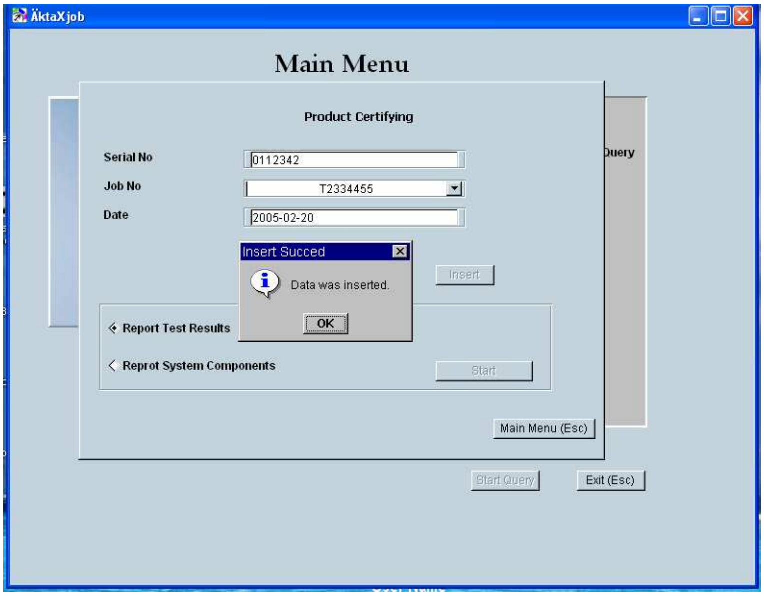
Figure 4.1: Archive System - Main menu




The main menu of the archiving system is shown in Figure 4.1. The main menu provide facilities to aid in inserting new jobs, inserting detail about the customer who orders the jobs, inserting details about products, details about the components and test carried out on products. The system also provides the ability to retrieve information like component details and test results about products. In addition reports on test results based on batch and product model can be obtained through the system

At the start of the manufacturing process of a product the details about the job number and serial number is inserted to the system. This is shown in Figure 4.2. Then the details of components built-into the products can be recorded during the manufactur-

Figure 4.2: Archive System - Details about the serial number and the job number are archived




ing process. The interface where one can do so is shown in Figure 4.3. The results of respective tests on the components and the completed product are inserted as shown Figure 4.4. This has to be done manually. But the test programs can also be developed to communicate directly with the database and thereby the need to manually insert data can be omitted.

When a product is completed, the information about the product and the results of the test, which vouch for the accurate functionally of the product, can be retrieved by the serial number of the product. The Figure 4.5 shows the interface to do that. The interface has the ability to take the whole serial number or the part of the number. When a part of a serial number is given a list of numbers is displayed as shown in Figure 4.6. Then the user is expected to choose the desired number and thereby will be able to view the record of the product as shown in the Figure 4.7.

It is vital to oversee the quality of the manufacturing processes to improve them. Different types of statistical reports are used to understand the quality of the manufacturing processes and the quality of components used. The Figures 4.8 and 4.9 are the summaries of the results of a certain test based on the product type and batch number respectively. The details about the test identity and the product type or batch number are provided to the system to generate these reports. These reports also provide the details about the maximum, minimum and average values of the test results.

Figure 4.3: Archive System - Details about the components included in the products are archived

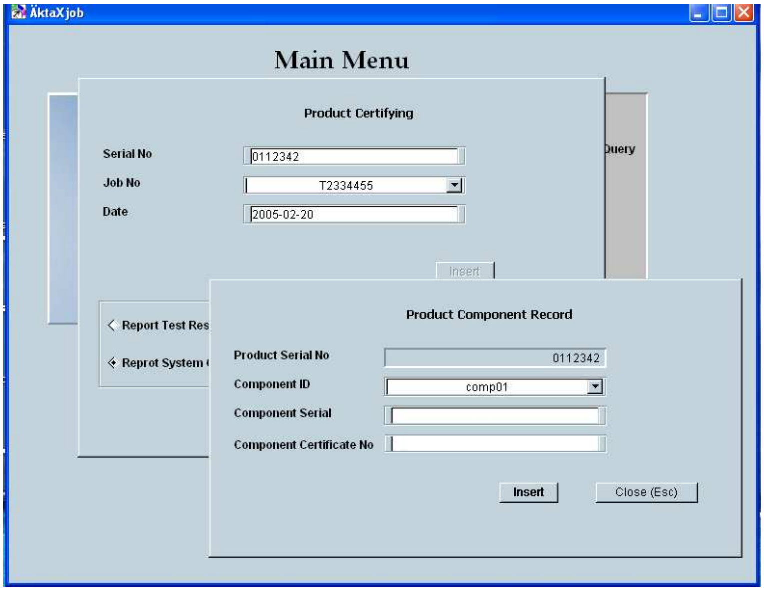



Figure 4.4: Archive System - The test results of each test carried out on a product are archived

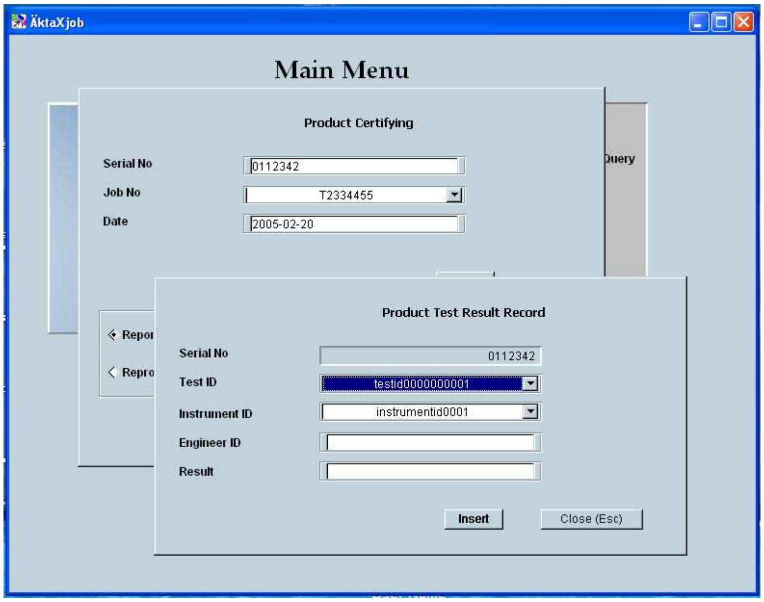


Figure 4.5: Archive System - Querying about product details with serial number of a product

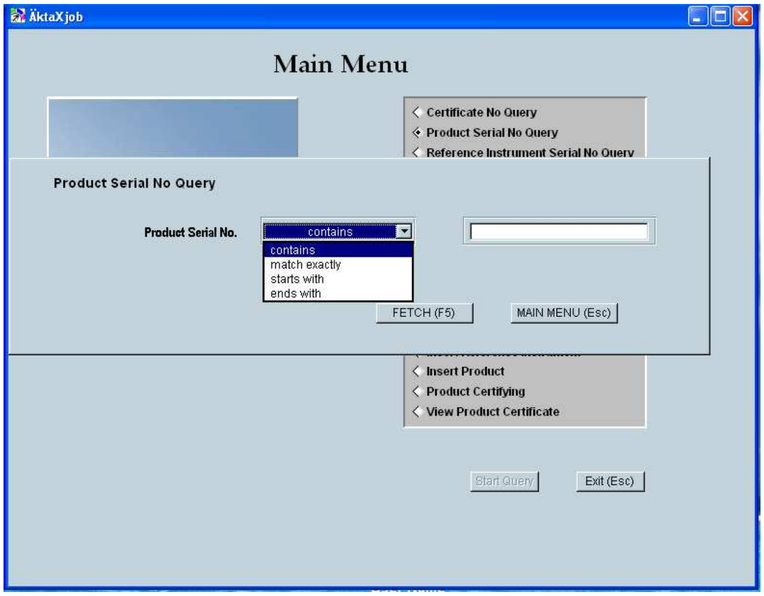



Figure 4.6: Archive System - A list of all the serial numbers is presented. Then a desired serial number could be chosen

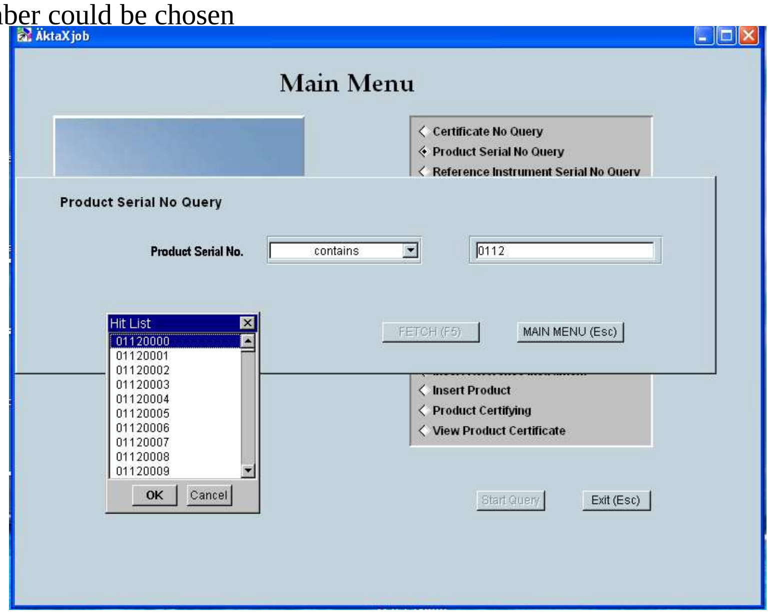


Figure 4.7: Archive System - When a serial number is chosen then the details of the product are presented

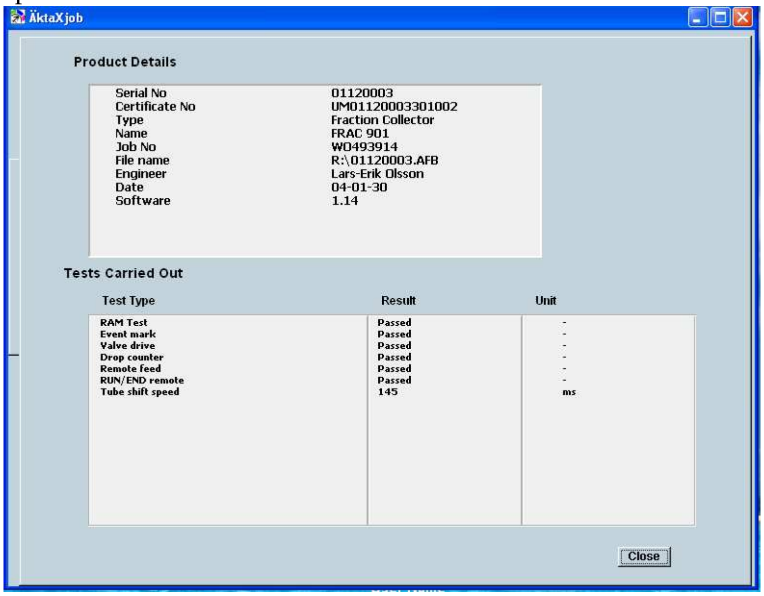



Figure 4.8: Archive System - The test results can be summarized based on product type and the test name

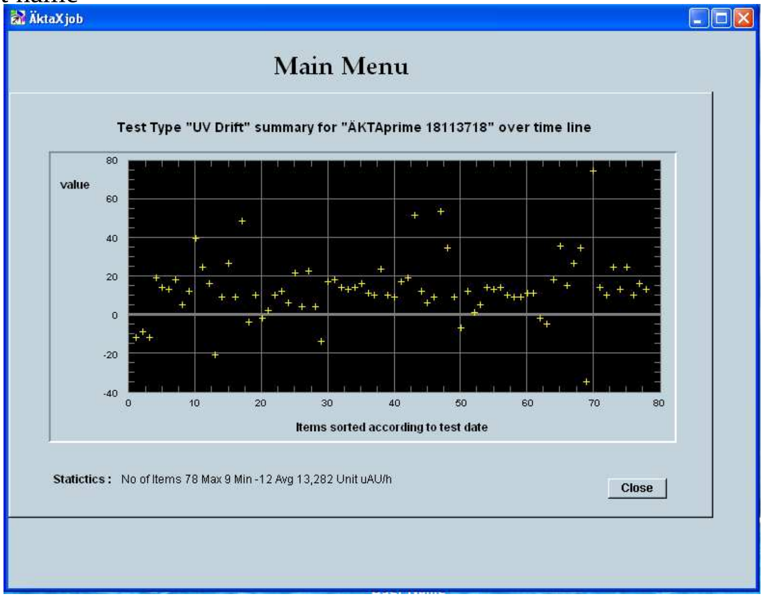
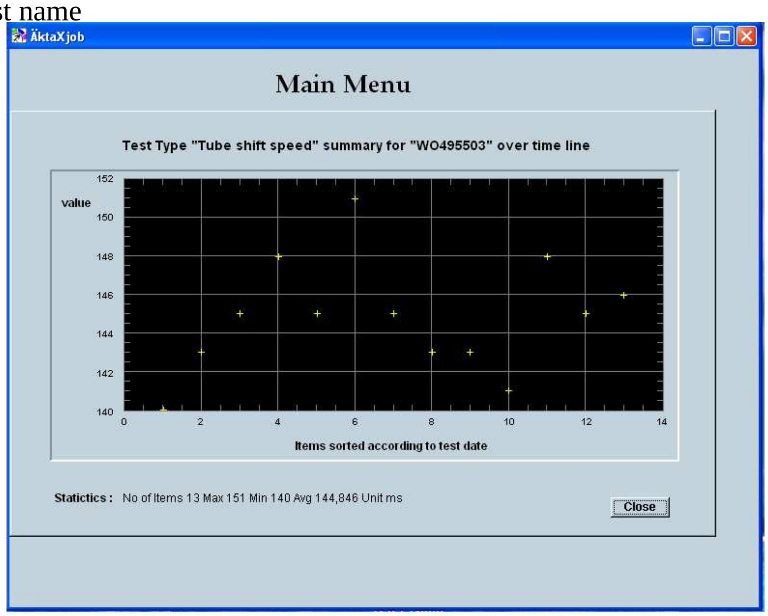


Figure 4.9: Archive System - The test result can be summarized based on job number and the test name




## 4.2 Diagnostic System

The user of the web based diagnostic system is expected to provide the system with the model of the product and the test identity of the failed test to start the diagnostic process as shown in Figure 4.10. Then the system would ask several questions about the situation and the user is expected to answer as shown in Figure 4.11. The Figure 4.12 shows the diagnosis derived by the system according to the input of the user during the session.

Figure 4.10: Diagnostic System - Choosing the product type and the test identity

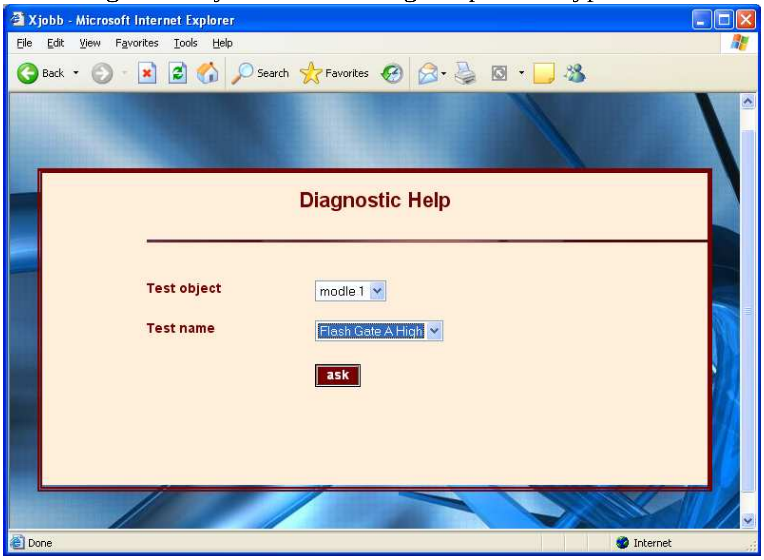



Figure 4.11: Diagnostic System - A question about the situation

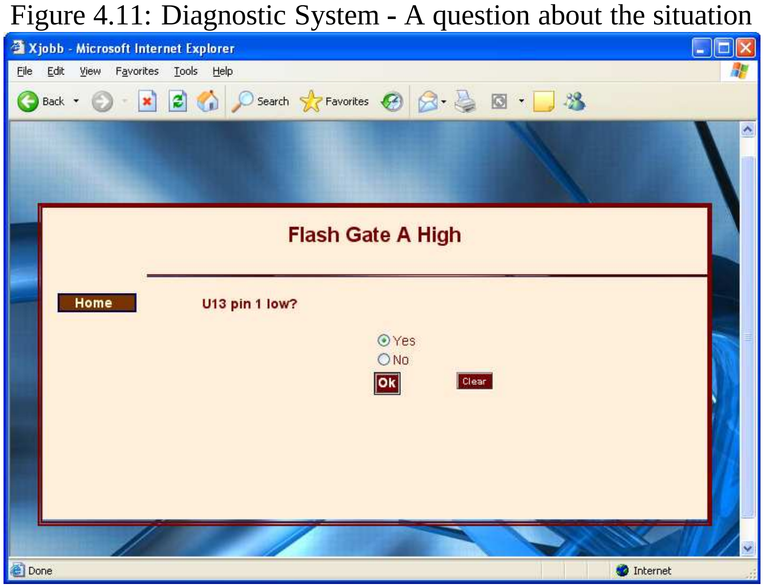
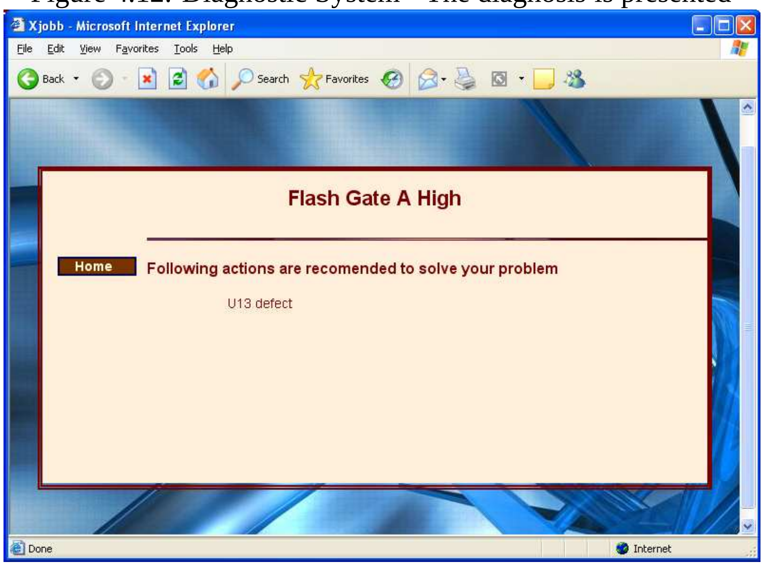


Figure 4.12: Diagnostic System - The diagnosis is presented



# Chapter 5

# Discussion

In this thesis concepts of rule-based reasoning, case-based reasoning and model-based reasoning have been presented in Sections 2.2.1, 2.2.2 and 2.2.3 respectively. It is essential to understand the strengths and weaknesses to use them wisely in different situations.

## 5.1 Rule-based Reasoning

Rule-based reasoning is a suitable method to use with the domains where a great deal of heuristics is used to solve problems. It also provides ability to debug conclusions derived. This is possible because of the fact that the rules are mapped in to the state space. The separation between the knowledge and the control offers the facility of developing the system iteratively. The system based on this approach is good for narrow domains where a lot of knowledge is not required to make an intelligent conclusion. The conclusions derived can be explained in the aspects of why and how.

A major weakness in this approach is that the rules that are derived from the human experts do not reflect the functional and model-based knowledge of the domain. The rules are incapable of handling missing information to solve problems. The explanations that can be provided with the mechanism are not based on the theoretical knowledge from the domain but based on the heuristic rules in the knowledge base. The knowledge base is task dependent in this approach which make the ability to cope with new problems weak. [4]

## 5.2 Case-based Reasoning

The main advantage in this approach is that the knowledge base can be developed using the case histories in the domain, repair logs which makes knowledge acquisition process easier. When a suitable case is found from the case storage finding a solution to the current case is faster than to generating a solution using rules and models. The past success and failures are taken to consideration in this approach. Complex situations can be coped with this reasoning because the system is learning form its past encounters. The management of the cases is easy in this mechanism. New cases can be simply added to the knowledge base. The requirements to manage the knowledge base are a suitable representation of cases, an effective retrieval index structure and a clever case alteration policy. The good indexing structure empowers the system.

As fore the disadvantage of case-based reasoning, the approach lack the ability to explain the conclusion deriving mechanism used during problem solving and deeper knowledge of the domain. The large knowledge base containing cases has to deal with the complications with storage/computing trade-offs. The difficulty generating of efficient indexing structures and similarity matching strategies make the advantages of the approach undeliverable. [4]

## 5.3 Model-based Reasoning

The key strength of this approach is the ability to represent the functional and structural knowledge of the domain in the knowledge base. Therefore, the systems based on this approach possess the ability to cope with new problems which were not taken into consideration during the initial designing process. This approach is considered to be robust, thorough and flexible because it falls back to first principles as a human would do in the case of new problem. The capability to provide with causal explanation is also another major advantage. This ability provides the users with greater understating of the problem situation.

This approach is unable to represent heuristics that the human experts possess. Furthermore, to use this approach, the underlying domain should have scientific theories or models describing the domain. The domains that lack such models can not use this mechanism successfully. The systems based on this approach are considered to be complicated and the inserting knowledge about special situations would result in unpredictable behaviors. [4]

## 5.4 Archiving System

The implementation described in Section 3.1 has the capacity of managing and archiving the data needed to create the necessary documentation to provide a better quality assurance. Additionally the advantage of this data stored in a database, is that the data can be used to create different reports to obtain deeper and broader views of the production. For example, reports about a given product, given component, given test program results, given product type, given component type, given customer and given supplier can be generated from the data in this database. Furthermore, by developing this system to save the failed test detail and possible cause to failure, would provide a better understanding about what are the common failures. This could lead to identify the cause of the product failure which could be for example a batch of defective components inside the product or an incorrect procedure carried out on a product which could affect the product destructively. Additionally, the database could archive test program and documentation describing the program structure for future reference. Another aspect is that schemas and documentation about the product could also be stored in the database which would provide a better management of such information. A new table with details about engineers could be added to the database instead of just recording the identity of the engineers who are involved in different procedures.

## 5.5 Diagnosis System

The current prototype is working as expected for the scenarios that it was developed for. A limitation in this implementation is that the knowledge base management requires knowledge about the implementation of the system. Therefore the system could be added with the facility to manage knowledge base with a knowledge base editor. This would provide a more user friendly interface to knowledge base management. Furthermore, the system could be developed for more complex scenarios which requires a Bayesian network to represent. The prototype's inference engine could be implemented with the help of stored procedures given that the database has facilities for that. This might improve the performance of the system equipped with a larger knowledge base.

# Chapter 6

# Conclusion

In my opinion the archiving system described in this thesis is fully capable of meeting the requirements of Amersham Biosciences AB. Implementing this system would provide a higher level of quality assurance for the customers and make it easier for the staff to maintain the documents involved in manufacturing processes. Further the system could be developed to generate new reports given they are based on the data currently included in the database. Oracle appears to be suitable as the database management system. Although Agilent VEE Pro can be used to create the graphical interface to the database, the choice could have been a more advanced programming language like Java. Such a choice would provide a more user friendly graphical interface. Another option is to provide a web interface to the archiving system using a dynamic web programming language like PHP. A different approach is a web application based on J2EE framework. In general using a database based product details archiving system is very suitable for the company's needs and requirements.

The diagnosis system is capable of handling the diagnosis scenarios considered during the design phase. The system is capable of handling similar diagnosis scenarios and it could be developed into more functional and usable one when the required knowledge base is established.

# Acknowledgments


- Dr. Michael Minock for his guidance during my thesis work.
- Mr. Jan Johansson, for giving me the opportunity to carry out my thesis work at Amersham Biosciences AB.
- Mr. Roland Vikström at Amersham Biosciences AB for supervising the market survey and providing all the necessary information about the manufacturing process carried out at the company.
- Mr. Karl-Erik Öberg at Amersham Biosciences AB for providing necessary information about the diagnosis scenarios.
- Mr. Mats Byström at Amersham Biosciences AB for providing the necessary information about the currently executed product details archiving process at the company.
- Ms. Sigrid Lindholm and Mr. Christian Häggström for proof reading and for their comments.
- My dear parents, brothers, aunts and friends for encouraging and helping me during my studies and thesis work.

# References

[1] Karthik Balakrishan and Vasant Honavar. Intelligent diagnosis system. *Aritficial Intelligence Research Group*, 2003.

[2] Robert Chi and Melody Kiang. An integrated approach of rule-based and case-based reasoning for decision support. *ACM Annual Computer Science Conference Proceedings of the 19th annual conference on Computer Science*, pages 255–267, 1991.

[3] Online Destiny Ltd. Agilent debuts visual engineering environment (vee) pro 7.0. http://www.embeddedstar.com/press/content/2004/3/embedded13408.html (visited 2005-02-10).

[4] George F Luger. *Artificial Intelligence Structures and Strategies for Complex Problem Solving*. Addision Wesley, fifth edition, 2005.

[5] Kevin Murphy. An intorduction to graphical models. http://www.cs.ubc.ca/~murphyk/Papers/intro_gm.pdf (visited 2005-02-10).

[6] John Mylopoulos. An overview of knowledge representation. *International Conference on Management of Data Proceedings of the 1980 workshop on Data abstraction, databases and conceptual modeling*, pages 5–12, 8 June 1980.

[7] Howard Shrobe Randall Davis and Peter Szolovits. What is a knowledge representation. *American Association for Artificial Intelligence*, pages 17–33, 1993.

[8] Stuart Russel and Peter Norvig. *Artificial Intelligence A Modern Approach*. Pearson Education, second edition, 2003.

[9] Thomas Schmidt. Model-based fault diagnosis. *International conference on Industrial and engineering applications of artificial intelligence and expert systems Proceedings of the second international conference on Industrial and engineering applications of artificial intelligence and expert systems*, 1:21–25, 1989.