Abstract

KOTCHER, PRATIK CHAMPALAL. An approach to develop knowledge representation for expert system to diagnose faults in domestic systems using qualitative modeling. (Under the direction of Dr. Edward Grant).

Faults, if remained undetected lead to huge financial losses. Troubleshooting faults requires analytical skills and knowledge of the system. Elementary skills help in solving common faults but faults that are infrequent or those which cannot be debugged by a consumer or even a field technician due to lack of in-depth knowledge of the system requires an expert's intervention. Experts are not always available in the field and lack of knowledge on the field technician's part may lead to huge financial losses and inconvenience on the part of the consumer. To avoid such situations an approach to make available expert's knowledge and capability to the field technicians is demonstrated through this research. A qualitative modeling approach to develop a knowledge representation is outlined here. A qualitative knowledge of the system from expert's point of view is modeled for a Bosch dishwasher for whom knowledge is extracted to generate a rule base, which reasons the characteristics of the system like an expert. Further it has been explored that using the rule base one can develop a questionnaire using the concept of model based diagnosis for troubleshooting. A qualitative model is useful for this kind of system because it resembles the analytical thinking capability of an expert and in general for any human being making it easier to communicate and interpret. Thus providing such a system to the field technician or even the consumer will simplify the troubleshooting process and will help in reducing downtime and financial losses.

An approach to develop knowledge representation for expert system to diagnose faults in domestic systems using qualitative modeling

By

Pratik C. Kotcher

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Electrical Engineering

Raleigh

2006

Approved By:

Dr. Edward Grant
Chair of Advisory Committee

Dr. Mark White
Member of Advisory Committee

Dr. Wesley Snyder
Member of Advisory Committee

DEDICATION

To my eldest sister
Late Dr. Shital Kotcher
whose presence I feel in each and every moment of my life

To my parents
for their constant support and belief in me

To my elder sister
Ms. Tejal Chavan
for encouraging me to reach my goals

BIOGRAPHY

Pratik Kotcher was born on May 27, 1981 in Mumbai, India. He received his elementary and higher secondary education from Don Bosco High School, Mumbai India. He received his Bachelor of Engineering from Thadomal Shahani Engineering College, Mumbai University in July 2004. In fall of 2004 he joined North Carolina State University to pursue his Master's in Electrical Engineering. He did a co-op with Bosch & Siemens Home Appliances Corporation, New Bern from May 2005 to December 2005. The experience and project he worked on during his co-op helped him to write his thesis.

Acknowledgements

I would like to express my sincere thanks to Mr. Bob Digman, Manager Dishwasher Department, Bosch & Siemens Home Appliances Corporation (B/S/H/ Corp) who provided me with the opportunity to work with him and apply the approach provided by Dr. Edward Grant, which led to this thesis. I am indebted to the staff of the Dishwasher Department at B/S/H for their constant support and especially to Matt Richards & Bob Jones, who provided me with the knowledge of the dishwasher system and the relevant data needed.

I thank Dr. Edward Grant for being my advisor and Chair of Committee, none of this research would have been possible without him. Thanks to Dr. Mark White and Dr. Wesley Snyder for agreeing to be members of my advisory committee and for their commitment to overseeing the thesis process. Finally, I thank my sister Ms. Tejal A. Chavan for proof reading and giving me space to work on my writing in her home.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 Introduction

Fault is a deviation from normal operation. It is an undesirable condition for a system which may result in machine downtime in industrial environment, leading to financial losses. Manuals of systems intended for consumer use at times are not adequate problem solving guides and may require field technician's intervention to restore the system. The field technician is generally someone in the field who has a relatively shallow understanding of the general technical principles compared to an expert (designers of the system), but usually more than the average consumer. Most system faults can be detected and restored by trained technicians quickly, but some faults require help from experts with a detailed knowledge of the system. This lack of expertise of the technician may cause the fault to reappear even if it may seem fixed at the time of repair. For example, when the field technicians of dishwasher department at Bosch & Siemens Home Appliances Corporation are unable to determine the exact cause of the breakdown of the system, they often change the Control module of the Dishwasher acting on their instincts, In some instances, it was found that the faults were due to loose connections between the components and were not detected by the field technicians due to lack of indepth system knowledge. This type of fault repair technique of the technician may lead to huge financial losses to both consumers and to the manufacturers if the breakdown occurs within the warranty period.

The analytical thinking process of an expert can be made available to the technician in the field through a tool known as an Expert System. This system will incorporate the decision making logic of a human expert. This research project outlines an approach to designing an expert system for fault diagnosis of Bosch & Siemens dishwasher. The approach outlined here consists of four major parts: 1) Designing a qualitative model of the dishwasher system, 2) Extracting the knowledge of the system, 3) Generating a rule base of the System, 4) Developing a questionnaire based on expert analysis for fault detection or the expert system shell. In this research report, a framework for only first three parts is designed and their result is presented. These three parts combine to form the knowledge representation in Expert System.

As mentioned by Forbus, K (1996) [1], a qualitative model is motivated by two observations. First, experts draw useful and subtle conclusions about the physical system without equations, working with far less data, and less precise data, than would be required to use traditional, purely quantitative methods. Second, they appear to use qualitative reasoning when initially understanding a problem, when setting up more formal methods to solve particular problems, and when interpreting the results of quantitative simulations, calculations, or measurements. Thus a qualitative model will help in developing a system with the reasoning capability of the expert. Developing a qualitative model is easier compared to developing a quantitative model, where qualitative messages (i.e. LOW, NORMAL, HIGH, ON, OFF) are passed from one component to another. A QuaMo Toolbox along with Matlab/Simulink [2] is used to develop the qualitative model of a dishwasher system. The QuaMo Toolbox provides the facility of developing simple linear models of the components of the dishwasher. Knowledge is extracted from the designed dishwasher and stored in an MS Excel format, which is available for generation of a Rule Base. The Rule Base is generated using WEKA (Waikato Environment for Knowledge Analysis). It was found that the Association algorithms provided with desirable Rule Base for the dishwasher system. The fourth and final part was not implemented, but tools for Expert System Shells like Visual Prolog ESTA, e2glite were explored.

The research is divided in following chapters; Chapter 2 gives a literature review by exploring the domains of qualitative reasoning, model based reasoning, expert systems, fault diagnosis methods, Chapter 3 discusses the qualitative design and implementation of the Bosch Dishwasher Model, Chapter 4 discusses the methods for the implementation of knowledge representation and provides with the experimental results for the same. Chapter 5 concludes the thesis and Chapter 6 provides with future directions on the work.

2 Literature Review

2.1 Artificial Intelligence

2.1.1 Qualitative Reasoning

This research/project describes an approach to develop a knowledge representation for an Expert System. The literature review explores the avenues of Artificial Intelligence (AI) investigating qualitative reasoning and explains the need for a reasoning tool which can qualitatively model a system. A qualitative model captures the causal structure of the system in a more profound manner than the conventional expert system and is not as rigid in nature as the numeric simulation which led to the development of many methodologies. These methodologies qualitatively represent knowledge and their reasoning.[4] qualitative reasoning is the area of artificial intelligence (AI) which creates representations for the continuous aspects of real world scenarios, such as space, time, and quantity, that support reasoning with very little information. Forbus, K (1996) [1] claims that typically, qualitative reasoning has focused on scientific and engineering domains, hence its other name, Qualitative Physics. Any physical work that human beings do in their everyday life has its origin in their imagination which is then manifested into reality in the form of machines. Qualitative Physics arises from the need to share human intuitive thinking about the physical world of the machines, representing and reasoning them about the physical world in qualitative sense. Its goal is to capture both the common sense of the layperson and the implied knowledge applied by the engineers and scientists. We must understand the foundation of qualitative physics by developing programs that capture these expertise. Lunze et al (1997) [2] have developed a QuaMo (Qualitative Modeling) Toolbox in

Matlab/Simulink which specifically aids the developing of qualitative models of physical systems.

Forbus points that [6], the essence of Qualitative Reasoning is quantization, which gives abstract symbols to the quantized states but also cautions that abstraction can be two-edged in defining correct spaces to the symbols especially while defining the boundaries between the two symbols. He also characterizes the styles of reasoning such as qualitative simulation, interpreting measurements, planning comparative analysis and others. Thus developing domain independent, generic algorithms for reasoning will help in dealing with larger systems. The potential application can range from designing to diagnosis of a system. The state of the art for qualitative reasoning is divided into three areas, (1) qualitative dynamics, which represents cause of changes, (2) qualitative kinematics, which represents spatial reasoning or limits to reasoning, and styles of reasoning which exploits the knowledge.

2.1.2 Model Based Reasoning for Diagnosis

Davis, R & Hamscher, W (1988) [7] discusses how Model Based Reasoning can be useful in cases of troubleshooting, they point out that to determine why something has stopped working, it is useful to know how it was supposed to work in the first place. It reasons a model of structure and behavior of the state of a device that the system is designed to simulate. It refers to inference method used in expert systems based on a model of the physical world. For troubleshooting, the model of the system or the predicted behavior of the system is compared with the observed behavior of the system. If there is a difference it is called as a discrepancy indicating a defect. This provides the starting point for diagnosis. An important part of the diagnostic ability of model-based reasoning is provided by behavior descriptions that capture both the casual behavior of the device (predicting outputs from inputs) and inferences that can be made about its behavior (determining inputs from outputs). The advantage of this technique is that there is no need to pre-enumerate what is going to be wrong, rather it identifies the discrepancy when the working of the device strays from its intended behavior. It is device independent and does not suffer from accumulation of experience, instead it reasons engineering principles applicable to wide variety of devices. Model based troubleshooting is symptom directed, it reasons from the observed misbehavior towards the underlying fault. Thus, in the system developed here we use a model base approach for generating the Rule Base and the Rule Base is not adaptive to experiences. The Rule Base consists of what is being designed in the qualitative model, see Figure 1.

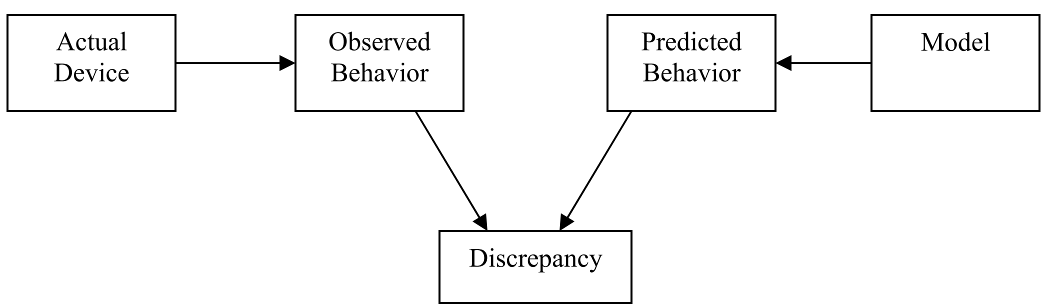


Figure 1 Model Based Reasoning for Troubleshooting

2.2 Expert Systems

An Expert System is a computer program that has an extensive knowledge base in a specific domain and uses an inference mechanism to draw conclusions like the human expert. [8] Classical Expert Systems [8] are based on symbolic reasoning but are often not capable of dealing with numerical processing. On the other hand, numerical algorithms and techniques can provide high precision to mathematically formulated problems, but are unable to provide insight into the problem solving process or interpretation of the results. Since both features, i.e. “generation” and “interpretation” of results are required in engineering applications; the current trend in engineering based expert systems (EBES) is to couple symbolic and numeric techniques. But for fault diagnosis rough information of system is enough to locate the fault, a qualitative approach with only symbolic representations is shown in this research. In general, Spyros Tzafestas (1993) [8], has classified it into three types: production rules which consists of single knowledge base and inference engine, structured production rules which has several knowledge bases using meta rules and distributed production rules which are hierarchal structure of network with co-operating specialists. The applications of Expert Systems covers wide range for solving problems in areas of interpretation, prediction, diagnosis, design, planning, monitoring, debugging, repair, instruction & control.

2.2.1 General Architecture

The general architecture of an expert system is shown in Figure 2. It can be divided in three parts as given by Spyros Tzafestas (1993) [8] which are:

(1) Inference Engine, the heart of the Expert System, acts as the solver of the problems, has the reasoning power of the expert and provides solution based on the facts given by the user.

(2) Knowledge Base, the memory of the expert system, consists of the rules and declarative description of the system.

(3) GUI or Working Memory, consists of the questionnaire for the end user, through whom the expert system obtains the facts and provides solution to the user. It is a communication link between the end user and the inference engine of the expert system.

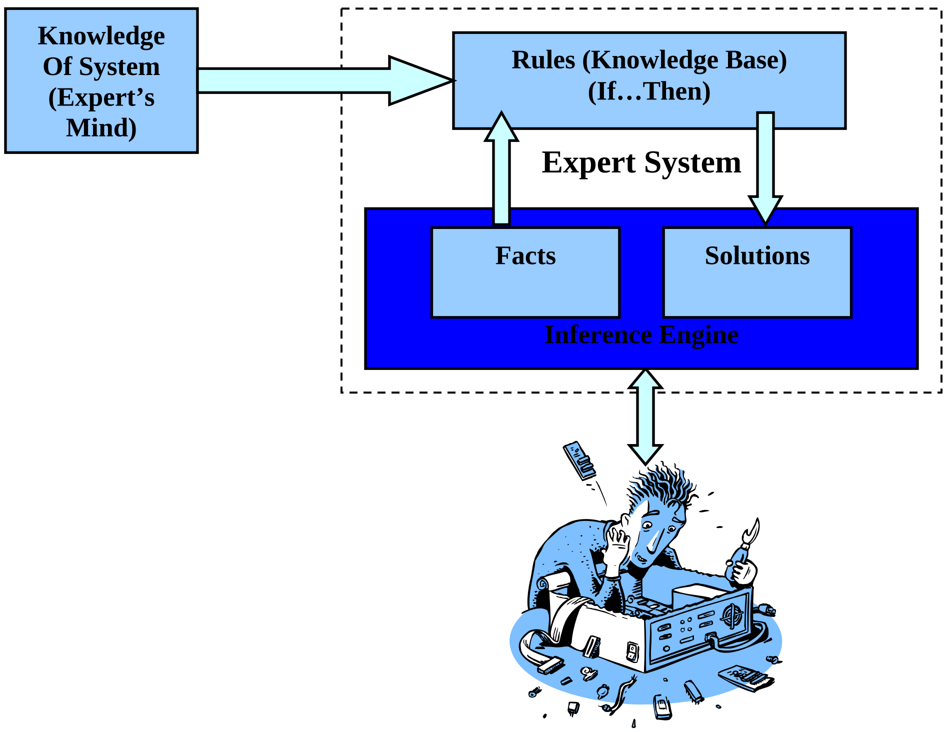


Figure 2 Expert System General Architecture

2.2.2 Knowledge Representation

Knowledge Representation is a method to convert the knowledge of a system into an applicable form for the expert system. Since the major emphasis of this research project is on developing a rule base, various techniques were explored for representing the knowledge. Spyros Tzafestas [8] has listed 6 different types of knowledge representation techniques.

(1) **Rule Based Systems** maybe viewed as 3 different parts: (i) a working memory stores facts, goals, and intermediate results, is similar to GUI or questionnaire, (ii) rule memory stores all the rules of the system, i.e. the knowledge base of the system and (iii) rule interpreter decides what rules are applicable and in which order they should be executed, i.e. the expert's mind or the inference engine.

(2) **Frame Based Systems** are data structures representing stereotyped situations.

(3) **Associative Networks** are network of nodes representing concepts and relations.

(4) **Logic Based Systems** are based on first-order predicate calculus.

(5) **Object Oriented Approach** have objects as data structures which besides being able to represent descriptive knowledge are also armed with private procedures or methods.

(6) **Attribute Grammar Representation,** a technique parsing mechanism and a semantic notation of attribute grammars that can be combined to represent the control knowledge and the knowledge base of logic programs respectively.

Spyros Tzafestas (1993) [8], also goes on to compare the rule based, frame based & logic based systems on the basis of efficiency and degree of difficulty in implementation. He claims that rule based systems are simplest to implement, logic based systems are fairly difficult and frame based systems are most difficult. Performance wise, he claims that rule based systems can be acceptable overall but deteriorate as the volume of knowledge base increases; frame based systems show no change in performance even if the volume of knowledge base increases; logic based systems are less efficient compared to other two systems and this is in accordance with intuition because resolution is a time consuming process. Finally, application domain and specific implementation is an important factor in choosing the right knowledge representation. A rule based method is implemented in this research due to its ease of implementation and to the availability of the supporting tools needed for development.

2.3 Fault Diagnosis Methods – General

Fault detection and diagnosis is an important problem in engineering. There is an abundance of literature on fault diagnosis ranging from analytical methods to artificial intelligence and statistical approaches. From a modeling perspective, there are methods that require accurate process models, semi-quantitative models, or qualitative models. At the other end of the spectrum, there are methods that do not assume any form of model information and rely only on historic process data. In addition, given the process knowledge, there are different search techniques that can be applied to perform diagnosis. Such a collection of bewildering array of methodologies and alternatives often poses a difficult challenge to any aspirant who is not a specialist in these techniques. Venkatasubramanian et al (2003), [3, 4, 5] have provided a systematic and comparative study of various diagnostic methods from different perspectives. A broad classification and review of fault diagnosis methods into three general categories is provided. They are quantitative model-based methods, qualitative model-based methods, and process history based methods.

2.3.1 Quantitative Methods

Residual generation through diagnostic observers, parity relation & Kalman filter are different types of quantitative model based approaches to fault diagnosis discussed by Venkatasubramanian et al (2003) [3]. It can be seen that one of the major advantages of using the quantitative model-based approach is that we will have some control over the behavior of the residuals. However, several factors such as system complexity, high dimensionality, process nonlinearity and/or lack of good data often render it very difficult even impractical, to develop an accurate mathematical model for the system. This, of course, limits the usefulness of this approach in real industrial system or processes.

2.3.2 Qualitative methods

Qualitative models are usually developed based on some fundamental understanding of the physics of the system. As discussed by the Venkatasubramanian et al (2003) [4] the qualitative models can be developed either as qualitative causal models or abstraction hierarchies. Digraphs, Fault Trees and Qualitative Physics are different techniques of causal models whereas structural hierarchy and functional hierarchy are the types of abstraction hierarchies. In terms of search strategies, they have broadly classified as topographic and symptomatic search techniques. Topographic searches perform malfunction analysis using a template of normal operation, whereas, symptomatic searches look for symptoms to direct the search to the fault location.

Though qualitative models have a number of advantages, the major disadvantage is the generation of spurious solutions. A considerable amount of work has been done in the reduction of the number of spurious solutions while reasoning with qualitative models. In Signed DiGraphs (SDGs), [4] this is done using generation of latent constraints and similar techniques have been proposed for Qualitative physics based models such as QSIM [4]. The search strategies were classified as either topographic or symptomatic search and the difference between these two types of search strategies is highlighted by them. Clearly, for a given qualitative representation, different search strategies could be used for diagnosis. Hence, one can view the methods proposed in the literature as different combinations of the qualitative methods and search strategies.

2.3.3 Process History based Methods

In contrast to the model-based approaches where "a priori" knowledge (either quantitative or qualitative) about the process is needed, in process history based methods, only the availability of large amount of historical process data is needed. There are different ways in which this data can be transformed and presented as "a priori" knowledge to a diagnostic system. This is known as feature extraction. This extraction process can be either qualitative or quantitative in nature. Two of the major methods that extract qualitative history information are the expert systems and trend modeling methods. Methods that extract quantitative information can be broadly classified as non-statistical or statistical methods. Neural networks are an important class of non-statistical classifiers. Principal component analysis (PCA), partial least squares (PLS) and statistical pattern classifiers form a major component of statistical feature extraction methods. The general limitation of process history based methods is not in the classifiers that are available to them, but in the availability of only a finite sampling of the distribution of the class data in the measurement space.

Venkatasubramaniam et al, (2003) [5] have compared and evaluated the various methodologies reviewed. This comparative study identifies the relative strengths and weaknesses of the different approaches. It also reveals that no single method has all the desirable features we stipulated for a diagnostic system. It is our view that some of these methods can complement one another resulting in better diagnostic systems. Integrating these complementary features is one way to develop hybrid methods that could overcome the limitations of individual solution strategies.

3 Dishwasher Model

3.1 Blocks of Dishwasher

Here is a brief description of different blocks of the dishwasher specifically for Bosch dishwasher model number SHX56C 02.

Control & Display Module: This is the controller of the dishwasher system with a microcontroller that controls the on-board relays to turn on-off various components of the dishwasher according to the program flow. Also it has an LED display, for displaying the current status of the dishwasher.

On/Off Switch: Switch controlling the main power supply 120V/60 Hz to the dishwasher.

Door Switch: A switch that disables all the components of the dishwasher except the control and display module when the door is closed. This safety feature prevents the dishwasher to working when the door is open.

Float Switch: A safety feature that shuts of the Main Pump and switches-on the Drain Pump in the event of water leaking from the Sump in the Base of the dishwasher

Water Inlet Solenoid: This controls the flow rate of water into the dishwasher depending on the source pressure. It switches on-off the source water supply to the dishwasher. The rate of water flow into the dishwasher is controlled by applying the natural pressure of the source water onto a diaphragm which modulates the water outlet depending on the source water pressure. Another feature of the Water Inlet Solenoid is that it acts as a filter to remove impurities from source water.

Flow Switch: This switch checks whether the flow of water is maintained inside the Flow Water Heater Assembly during the normal operation of the dishwasher.

Heating Element: This is the heater inside the dishwasher system, it raises the water temperature to the required temperature needed for cleaning.

Thermostat + NTC: The NTC (Negative Temperature Co-efficient) unit helps in keeping the Heating Element temperature within set limit. The thermostat acts as a temperature fuse to switch-off the supply to the Heating Element if the temperature rises above a set maximum limit.

Circulation Motor or Main Pump: This unit generates the required pressure to rotate the spray arms inside the dishwasher**.** It simply propels the water from the Sump with the help of the impeller and generates the force to rotate the spray arm to cover the entire area of the dishwasher.

Dispenser Actuator: This is a motor that opens the dispenser door when actuated in the required cycle to dispense the washing detergent into the tub for cleaning.

Drain Pump: The drain pump drains water out of Sump whenever the cycle demands or when the safety feature is invoked to empty the water out of dishwasher.

Aqua Sensor: This sensor determines the turbidity of water and drains off water with high level of turbidity. It helps in better cleaning, as clean water is introduced whenever the level of turbidity in water is high.

Rinse-Aid Sensor: It detects the level of the Rinse-Aid in the Rinse-Aid unit.

Water Switch: This switch allows only the top rack spray arm to function when selected.

Info Light: This feature is not available in all the models. It indicates whether the dishwasher is working in its cleaning and/or washing cycle.

Water Level Switch: This safety feature checks whether the rate of water filling the Sump of the dishwasher is within the safety limits. There are generally two types of switches, Timed Filled & Pressure Filled. Timed Filled switch checks the flow rate of water into the Sump and if it is above the limit then Water Level Switch is activated. Similarly, Pressure Filled switch is not used in this model, but this switch is activated if the water pressure is too high.

3.2 Qualitative Description of Dishwasher Model

A qualitative block diagram for the dishwasher is shown in figure 1. This model was developed to aid in designing the qualitative model of the dishwasher in the QuaMo Toolbox of Matlab/Simulink. Among the various models of dishwasher being produced by B/S/H/, the given qualitative model is developed specifically for SHX56C 02.

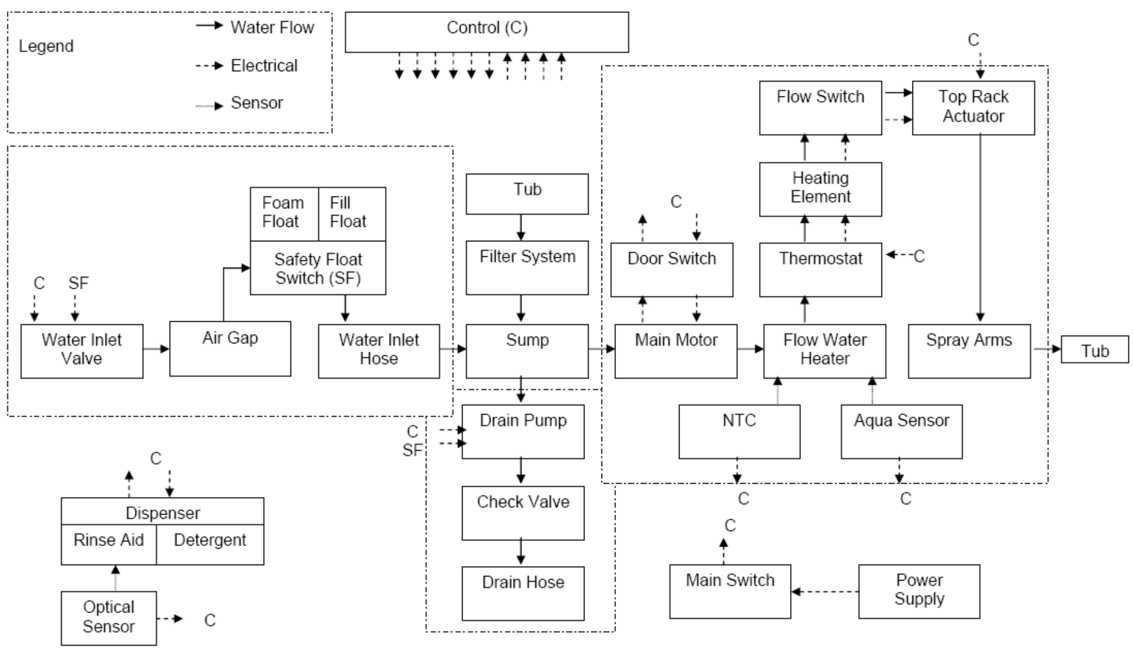


Figure 3 Qualitative Block Diagram of Dishwasher

The advantage of the qualitative model is that it refers to symbolic values unlike numeric values in quantitative models. These symbolic values can resemble the human knowledge of reasoning with symbolic messages passed from one block to another, thus simulating the process of human thinking for a particular problem or in this case a system. Figure 1 shows a simple qualitative model of the dishwasher system with its most important components.

Here is a brief explanation of the design of the qualitative model of the dishwasher, and explaining the requirement for a qualitative model. A qualitative model does not require any complex mathematical equations and quantitative values for finding solutions; it simply requires abstracted input/output values partitioned over its minimum and maximum range. Some examples of partitioned qualitative input/output are {Low, Medium, High}, {On, Off}. In fact, the whole system can be designed using these sets of values by referencing to the physical quantity of each component of the system with the given qualitative values in desired partition. The overall system consists of passing one of these partitioned qualitative messages from one block or component to another thus producing the desired result in the form of intuitive understanding for the system using qualitative symbols.

In the qualitative dishwasher system model designed here, it has three paths for passing messages, (1) the path of flow of water through the system, (2) is the electrical or the control path, which commands each blocks and (3) the sensor path. For the component, each block QuaMo toolbox is a simple linear State Space model. Each component of the dishwasher has a partitioned input, output and state space, the detail of which is explained in QuaMo Toolbox section in Appendix I:

Control: Although this block is not a separate entity at this design stage, its functions are present all over the system. For ease of simulation, the block is not centralized but in future it can be converted into a separate centralized unit. In general, this block provides control signal to other blocks, thus enabling/disabling them depending on the algorithm, which represents the partitioned qualitative signals {On, Off}. In most cases the control algorithm is time based, but in some cases the decision making process is carried out by the control with the help of feedback it receives from the sensors.

Water Inlet Valve: This block has its input, state and output space partitioned as {Low, Medium, High}. The input and output are based on the flow rate. The input is defined from the source water supply with one of the three partitioned states of the qualitative input space. The state space model of the water inlet valve is defined by the linear state space co-efficient. This block should be controlled by the Control depending on the algorithm; also it is controlled by the safety float asynchronously when a leakage is detected. In the current model the Safety Float feature is not implemented yet. The output is directly given to the Sump, eliminating the Air-gap and Water Inlet hose for simplicity.

Sump: The input space of the Sump is the flow rate, which is same as the output space of the Water Inlet Valve. The state space of the Sump is defined by volume and the output space is the level of water in the Sump. The Water Inlet Valve pours water into the Sump to the required level and this water is circulated by the Main Pump into the Tub and back into the Sump.

Main Pump: This component is also known as the Main Motor or the Circulation Pump. Its input space has 2-D, inputs one from the control to turn it On/Off at desired time and other from the Sump indicating the level of water in the Sump. The state is the pressure generated by the impeller of the Main Pump and the output is the flow rate.

Flow Water Heater (FWH): This unit physically comprises of NTC (Negative Temperature Co-efficient) Sensor, Thermostat, Flow Switch, Heating Element, Aqua Sensor and Top Rack Actuator. The input, state and output state for it is defined by the same parameter i.e. Flow Rate. The input is taken from the Main Pump. The other units are generally sensors except the Heating Element and Top Rack Actuator which provides direct control of the processes whereas the sensors indirectly control the processes. The qualitative models of these units are explained subsequently.

NTC (Negative Temperature Co-efficient) Sensor: The NTC senses the temperature of the water so its input is Temperature and checks whether the temperature is in-range or cut-off and thus controls the process of switching the heating element On/Off based on the algorithm and temperature of water heated by the heating element. The implementation of this unit is by measuring the temperature on the Heating Element as it rises and changing the heating process based on the set-points. Basically, NTC sensor is not implemented as a separate qualitative model but its output is derived by checking the temperature value on the Heating Element. Its output is defined by {In-Range, Cut-Off}.

Thermostat: This unit is simply an extension of NTC Sensor but it extends itself by acting as a fuse for the whole system and cuts-off the power to the Heating Element if the temperature rises above sustainable limit. Although this unit has not yet been implemented, it will have a similar implementation as NTC when implemented.

Flow Switch: The implementation of this unit is not a separate qualitative unit but it detects the Flow Rate in the FWH and generates output from the set {On, Off} determining flow of water in the FWH.

Heating Element: The Heating Element is represented by an integral rise in temperature to supply heat to the flowing water inside the FWH. The input to the Heating Element is from the Control {On, Off}, the state is defined by the rise in temperature of the Heating Element and output is defined by detecting whether the Heating element is On/Off.

Note: Aqua Sensor and Top Rack Actuator are not yet implemented in the current qualitative model because of the complexity in their functioning.

Spray Arms: The input to the Spray Arms is the Flow Rate from the FWH; its state can be visualized as the pressure created by flowing water; and the output is the rotational motion representing it discretely by {Rotating, Not-Rotating}.

Dispenser: The Dispenser dispenses the detergent into the tub of the dishwasher at the instant given by the time based algorithm. Its input space is defined by the Control which commands it to dispense the detergent. The output space is discrete in nature simply representing whether it's in either {On, Off} state.

Rinse Aid: The design and function of Rinse Aid function quite similar to Dispenser, the algorithm separates the instant it needs to be in the output state of {On, Off}.

Drain Pump: The Drain Pump switches 'On' at a particular instant as defined in the algorithm, which enables to drain the water out of the tub the of dishwasher. Its input space is 2-D, a Control {On, Off} and Level of Sump as {Low, Normal, High}. The state of the Drain Pump is not implemented "as is" due to linear nature of the system but can be visualized as the pressure generated to drain the water out of the Tub of dishwasher. The output space is represented simply in discrete states {On, Off}. Apart from having a synchronous control in the control algorithm, the Drain Pump is switched "On" asynchronously as a safety feature by the Safety Float Switch.

3.3 Customer Service Program

3.3.1 Algorithm of Customer Service Program

The Qualitative model of the Dishwasher is based on the algorithm of the Customer Service Program (CSP) for the model SHX56C 02. The steps involved in this program are given in Table 1.

Table 1 Customer Service Program Algorithm [Reference from B/S/H/ Customer Service Program]

Step or Index	Function	Time (seconds)	Note
1	Drain Pump On	15	Drains any initial water from the sump before new water is supplied
2	Filling (Water Inlet Valve On)	36	The water starts filling into the sump in this step, the water fills at the rate of 2.5 L/min and it goes on to fill 1.5 L so this cycle goes on for about 36 seconds.
3	Break	5	Causes break for 5 seconds all components are on a hold
4	Main Pump + Filling (Water Inlet Valve)	72	In this step the Main Pump starts pumping water into the Tub; also the Water Inlet Valve is switched On again to supply 3L more of water to completely fill the sump.
5	Main Pump + Heating Element + Dispenser Detergent	120	In this step the actual washing starts with the Main Pump pumping the water into the Tub through the Spray Arms, the heater is turned on for heating the water and the Detergent is dispensed by the Dispenser into the Tub.
6	Main Pump + Heating element	1620	This is the main step and the longest, here the Heating Element heats the water to the desired temperature level of 60 degrees C. Since the rise in temperature is about 1.5 degrees C/min, the total time required is 27 minutes or 1620 seconds.

Table 1 (Continued)

7	Main Pump + Dispenser Rinse Aid	120	The Main Pump continues to pump water into the Tub through Spray Arms and the Rinse Aid is switched On dispensing some Rinse Aid liquid into the Tub.
8	Main Pump + Heating Element + Aqua Sensor	20	The main aim of this step is to calibrate the Aqua Sensor with respect to the clarity of water. If the water is dirty beyond this limit the water is drained and a new wash cycle starts in a normal wash cycle.
9	Sputter Draining	30	The draining of water from the sump after the washing is complete starts at this step. In this step the draining takes place intermittently, the Drain Pump is switched On for 5 seconds and switched Off for next 5 seconds. This cycle continues for 30 seconds.
10	Drain Pump	45	The draining continues with the Drain Pump switched On for 45 seconds continuously
11	Main Pump + Drain Pump	15	This step ensures that all the water is drained out of the sump.

3.3.2 Implementation of Customer Service Program Algorithm in QuaMo Toolbox

This section describes the implementation of the CSP algorithm in the QuaMo Toolbox in Matlab/Simulink. Each step of CSP described in Table 1 is scaled down into arbitrary time instances. The total time span of the CSP algorithm in Table 1 is scaled down to 40 instances and each step time is divided and scaled arbitrarily. The reason for arbitrary scaling and division of the time instant lies in that fact that the model is just a simulation of the actual model and the given simulation provides the relevant information needed for further processing and also to maintain simplicity.

Table 2 Implementation of Customer Service Program in QuaMo Toolbox Algorithm

Step or Index	**Instances in the Qualitative Model**	**Components Active**	**Note or Remarks**
1	1-2	Drain Pump	Step 1 is simulated by first 2 instances in time when the Drain Pump is switched On, all the other components are off.
2	3-5	Water Inlet Valve, *Sump*	Step 2 is simulated from the 3rd instant when the Water Inlet Valve is switched On and starts pouring water into the Sump, the water level in the Sump starts increasing from this step, the Drain Pump is Off in this step.
3	-	-	Not implemented
4	6-10	Water Inlet Valve, *Sump*, Main Pump, *Flow Switch*, *Spray Arms*	At the 6th instant the Main Pump is switched On, which starts Rotating Spray Arms. The Water Inlet Valve continues pouring water into the Sump till it is completely filled at the 10th instant.
5, 6	11-23	Main Pump, *Flow Switch*, *Spray Arms*, Heating Element, Dispenser Detergent	Step 5 signifies the start of main washing cycle starting at instant 11th the Heating Element is switched on and detergent is dispensed by the Dispenser Detergent. Step 6 cannot be actually distinguished from step 5 as the Dispenser Detergent although not shown in the previous step still seems to be On as the wax motor is switching itself to Dispenser Rinse Aid. Apart from this the Heating Element continues its integral rise in temperature till the NTC reaches the cut-off range and switches Off the Heating Element at 24th instant.

Table 2 (Continued)

7	24-30	Main Pump, *Spray Arms,* Dispenser Rinse Aid	Step 7 continues the wash cycle, Rinse Aid is dispensed in this step
8	-	Main Pump, Heating Element, Aqua Sensor	Not Implemented
9,10,11	31-40	Drain Pump, *Sump*	The last 3 steps are combined together to represent as a single drain cycle, where the water is completely drained out of the Sump. It is performed in the instances 31-40. The reason for keeping it long for in case of future additions or modification in the instances.

***Note:** The Components in italics are not directly controlled in a particular step but are affected or made active by other components.*

4 Experimental Results

4.1 Methods

The Qualitative model of the dishwasher system was build using QuaMo Toolbox in Matlab/Simulink. The qualitative knowledge or information of the dishwasher system is extracted from the Matlab/Simulink qualitative model and stored in a spreadsheet which is then converted to CSV (Comma Separated Value) file, which is subsequently formatted into ARFF (Attribute Relation File Format) required by WEKA (Data Mining/Machine Learning Tool) by adding the attribute definitions.

4.1.1 Qualitative Modeling Methods

The qualitative model of the dishwasher system developed in Matlab/Simulink can be divided in two parts. The first part consists of the back processing performed in Matlab (file with extension .m). The results generated from Matlab program is subsequently used by the second part, which is qualitative simulation performed in Simulink (file with extension .mdl). The Simulink simulation represents results of the dishwasher system simulated in the form of qualitative plots for the dishwasher components and displays the components representing the qualitative states of the dishwasher during the execution of the Customer Service Program.

The first part or the back processing is composed of qualitative description of each part of dishwasher. The qualitative description is standard and in the same format for each component. It can be decomposed in five parts as described below:

a) <u>State Space Model:</u> Linear state space model is used for each component. The state space model is described by its co-efficient for each component. For most components this model represents a simple input-output relationship.

b) <u>Partitioning:</u> The input, output and state parameters of each component is partitioned in qualitative states using their quantitative values. The partitioning can be simply done by representing each components quantitative parameter in the range of 0 -1.25. The partitioning is standard for most components, such as 0-0.25 defines the LOW range, 0.25 - 0.75 for NORMAL or MEDIIUM range, 0.75 -1 for HIGH range and 1 -1.25 for VERY HIGH range. The VERY HIGH range is generally used only in situations of overflow or out of range. For discrete component parameters, 0 represents OFF and 1 represents ON.

c) <u>Mesh-point Generation for Behavioral Relation:</u> The input and state space requires generating mesh-points in the area defined by its upper and lower limits. These mesh-points are related using the state space co-efficient and partition values to generate the behavioral relation.

d) <u>Behavioral Relation:</u> The behavioral relation of a stochastic automaton describes a conditional probability that the automaton moves from the current state (z) to the successor state (z') while giving the output (w) if it obtains the input (v). The behavioral relation thus generates a set of conditional probability for the output (w) and the successor state (w) for all points in the mesh-points generated. Thus the behavioral relation requires mesh-points, state space co-efficient and the partition space.

e) <u>Initial States:</u> An initial state in the form of probability vector using normal distribution is generated for each of the qualitative partition. Generally, each component is initialized in its normal state. [14]

Below is an example of the back processing code for the qualitative model of Water Inlet Valve, the same code is used for all components with variations in state space co-efficient, partition space and initial states.

```
%%%%%%%% Standard Back Processing Code for Each Component – Water Inlet Valve %%%%%%%%%

% a) Ideal State Space for a linear discrete time input=output relationship
A1=[0];
B1=[1];
C1=[0];
D1=[1];

% b) Partitions
parx_WIV = partition([0 0.25 0.75 1.0 1.25],'r');
pary_WIV = partition([0 0.25 0.75 1.0 1.25],'r');
paru_WIV = partition([0 0.25 0.75 1.0],'r');

%  Abstraction
%  ===========

%  c)  Numbers of meshpoints for approximation of L
area_xs_WIV = [100 0 1.25];
area_us_WIV = [100 0 1];
area_ps = [];

%  Abstraction options
FStoc = 1;                    % Force automaton to be stochastic
RunSilently = 0;              % Output during calculation
AbstrOpt=[FStoc RunSilently];

%  d) Compute behavioural relation (index format) of Water Inlet Valve
L_WIV= linss2a(area_xs_WIV,area_us_WIV,area_ps,AbstrOpt,A1,B1,C1,D1,parx_WIV,paru_WIV,pary_WIV);

L_WIV

% e) Initial State
x01_WIV = [0.5]';
pz0_WIV = ev2pv(x01_WIV,parx_WIV);
pz0_WIV

%%%%%%%%%%%%%%%%%%%%%%% END %%%%%%%%%%%%%%%%%%%%%%%%%%%
```

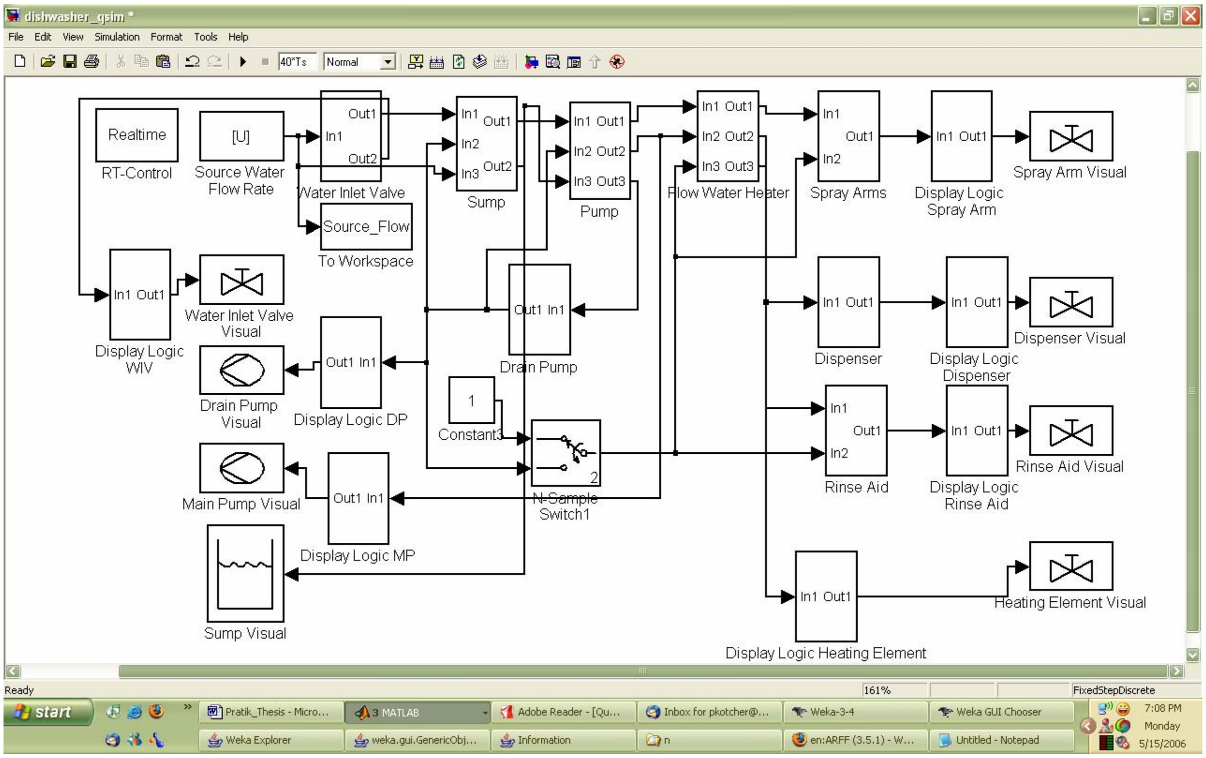


Figure 4 Qualitative Model of Dishwasher developed in Simulink

The second part of the dishwasher i.e. the qualitative simulation is shown is Figure 4. The blocks shown in the figure corresponds to the dishwasher components as shown in Figure 3. Each block contains the logic for the corresponding dishwasher component. It consists of three major blocks from the Qualitative Modeling Library (Qlib) of QuaMo Toolbox, which are used by every component. The other blocks are useful for data conversion from quantitative to qualitative and/or control blocks to control the flow of the algorithm. The important blocks are explained below: (For more details on other blocks see Appendix I)

a) <u>Probability Generator (PGen):</u> This block generates a probability vector output from the qualitative number input. The probability vector is normal distribution of the qualitative number. The probability vector is required by the Qualitative Simulator (Quasi) as its input. The input partition dimension is the required parameter for the probability generator.

b) Qualitative Simulator (Quasi): This block generates the real time output for the given input using the behavioral relation of the given dishwasher component. It requires a probability vector as its input and its parameters are behavioral relation, initial state distribution, and the sampling time. It provides output in qualitative number or probability vector which can be the current state or output of the component. A qualitative number is used when Quasi is connected to the Q-Scope.

c) Qualitative Scope (Q-Scope): This block displays the output results for each component of the dishwasher system qualitatively for each instant. The qualitative output highlights one of the ranges of output partition space of each component, representing the active qualitative state of the component.

Additionally, an interactive display is provided with the plots, which shows the states of the components in each instant when the simulation is executed. This display uses the blocks from the Process Supervision library (PSlib) of QuaMo Toolbox, which uses the qualitative number from Quasi Block output.

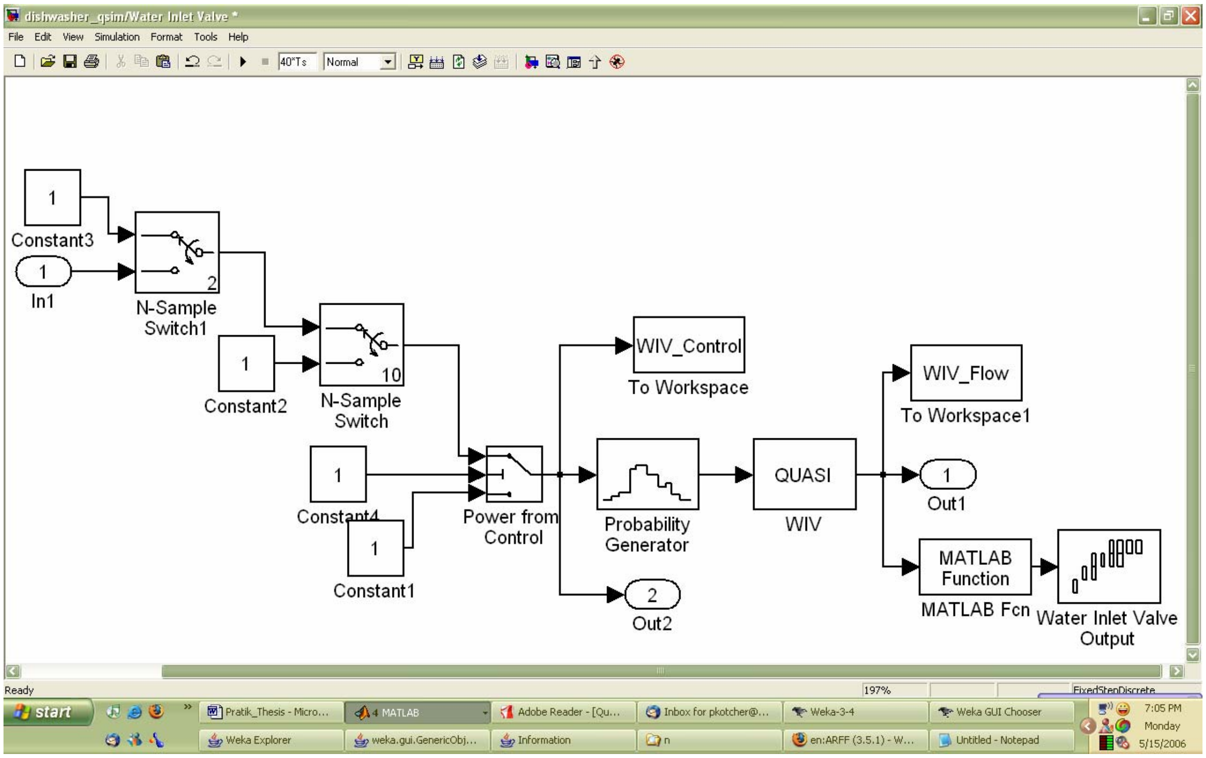


Figure 5 An Example showing the logic and QuaMo Blocks for Water Inlet Valve

4.1.2 Knowledge Extraction Methods

The qualitative model of the dishwasher system provides with information of normal working of the dishwasher according to the CSP algorithm. From this algorithm we know the different qualitative states of each component at a particular instant. This knowledge is in numerical form corresponding to the qualitative state. For example, '1' corresponds to a qualitative value of LOW, which is similar to enumeration data type in C-language. The numerical values are stored in a variable in the Matlab/Simulink workspace. After the simulation is completed the Simulink callback function known as Stopfcn () is executed, which includes the function (save_data.m) to write the results from the stored variables in the workspace to a spreadsheet. The function also converts the numerical qualitative results to symbolic representations (e.g. LOW, NORMAL, HIGH, ON, OFF).

The knowledge is extracted for different attributes such as SOURCE_FLOW, WIV_FLOW and so on which represents the states for different components as described in its model. Thus a knowledge base is formed of the dishwasher system for the CSP algorithm for each instance (40 instances as explained in the Implementation of CSP).

4.1.3 Knowledge or Rule Base Methods

The Knowledge or Rule Base of the dishwasher system was generated using WEKA a collection of machine learning algorithms for data mining tasks. WEKA was mainly targeted towards generating rules for the dishwasher system using association rule algorithms rather than providing prediction of faults for the dishwasher system.

The Rule Base is developed from the extracted knowledge of the system stored in the spreadsheet (with extension .xls). This extracted knowledge needs to be converted into ARFF (Attribute Relation File-Format) required by WEKA. For more details on ARFF, refer the Appendix III. The conversion process from spreadsheet to ARFF is done using following steps:

a) The MS Excel spreadsheet is converted to CSV (Common Separated Value) format (with extension .csv). This is done by saving the file type as .csv (Comma Delimited) in MS Excel itself, the filename can be same with no changes made to the original MS Excel file. This makes available the data set in any word processing or editor environment with the column separator now substituted by comma separator as required by ARFF.

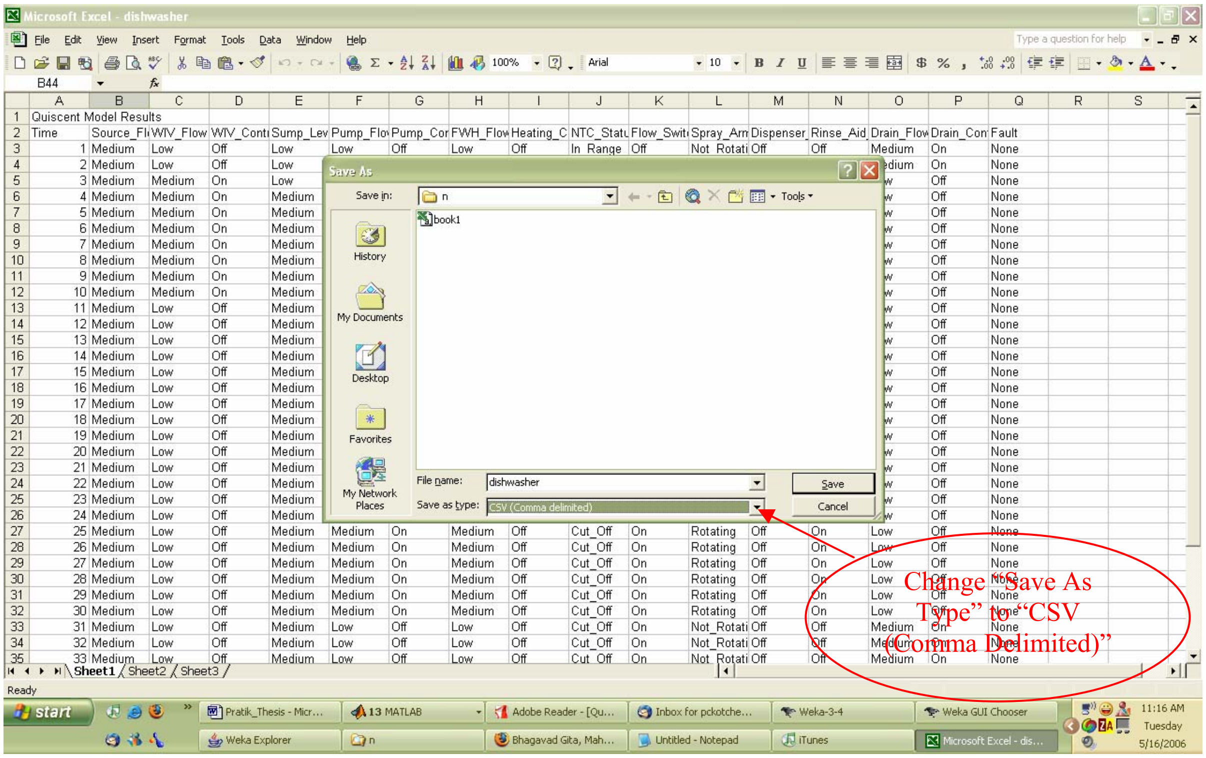
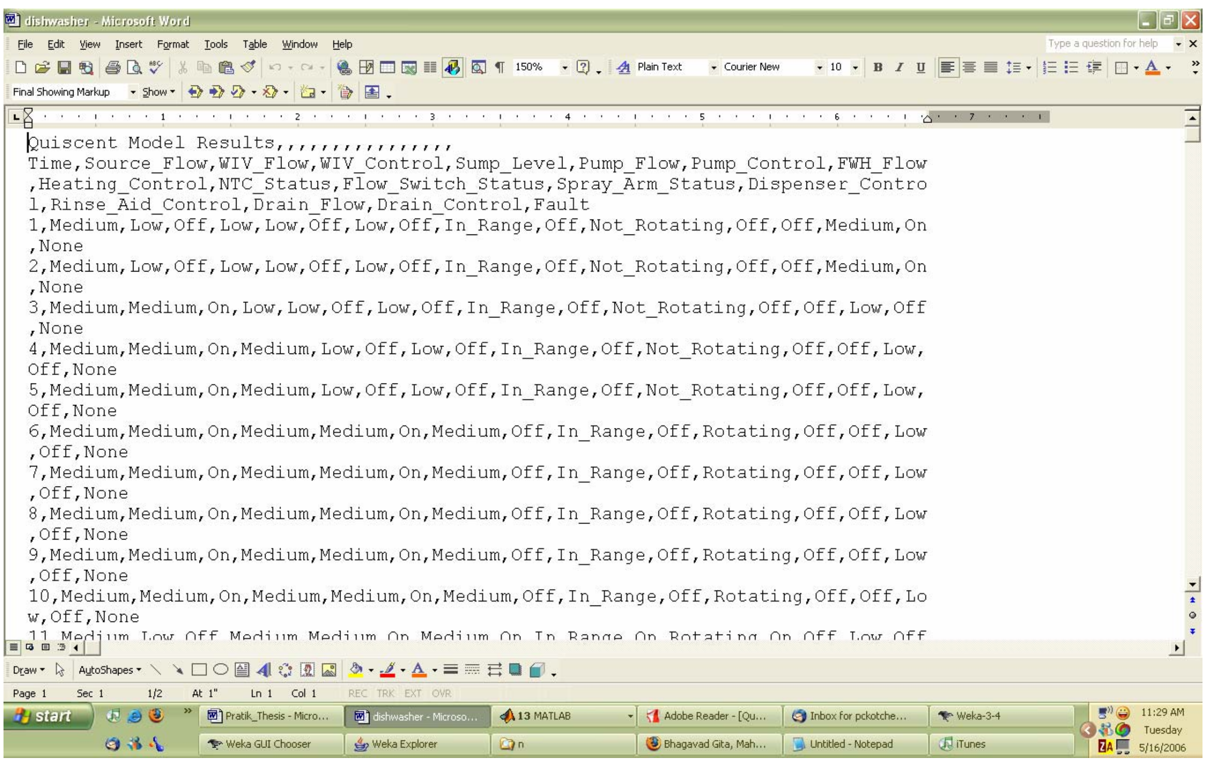


Figure 6 Converting from MS Excel (.xls) format to CSV format

b) Now the conversion to ARFF from CSV is simplified as CSV will separate the attributes in each instant by a comma unlike a column separator in MS Excel as required by ARFF. Next the rules or syntax required by ARFF is added using an editor. The required rules or syntax are to define the name of relation, different attributes present in the relation and finally its data set. Relation is the name given to the dataset and is defined by the command @RELATION <relation-name> (e.g. @RELATION dishwasher). Attributes are the column names which defines in most cases the output of the components of the dishwasher. The attributes are related to each other and WEKA helps in finding this relationship between different attributes. Attributes are defined by the command @ATTRIBUTE <attribute-name> <data-type> (e.g. @ATTRIBUTE Source_Flow Nominal Specification). The data-set we extracted from the qualitative dishwasher model with all attributes representing Nominal data-types with the qualitative values such as LOW, NORMAL, HIGH, ON, OFF and so on. In the set of attributes, if a class attribute is not defined WEKA automatically considers the last attribute as the class during application of any algorithm; the class can also be specified during application of the algorithm in WEKA. Final part of formatting into ARFF is to add the command @DATA at the start of the data-set separated by commas and thus making the data set available for WEKA. Finally, save the file with .ARFF extension (i.e. filename.ARFF) and select its type as Plain Text if the editor is MS Word.



```
Quiscent Model Results,,,,,,,,,,,,,,,
Time,Source_Flow,WIV_Flow,WIV_Control,Sump_Level,Pump_Flow,Pump_Control,FWH_Flow
,Heating_Control,NTC_Status,Flow_Switch_Status,Spray_Arm_Status,Dispenser_Contro
l,Rinse_Aid_Control,Drain_Flow,Drain_Control,Fault
1,Medium,Low,Off,Low,Low,Off,Low,Off,In_Range,Off,Not_Rotating,Off,Off,Medium,On
,None
2,Medium,Low,Off,Low,Low,Off,Low,Off,In_Range,Off,Not_Rotating,Off,Off,Medium,On
,None
3,Medium,Medium,On,Low,Low,Off,Low,Off,In_Range,Off,Not_Rotating,Off,Off,Low,Off
,None
4,Medium,Medium,On,Medium,Low,Off,Low,Off,In_Range,Off,Not_Rotating,Off,Off,Low,
Off,None
5,Medium,Medium,On,Medium,Low,Off,Low,Off,In_Range,Off,Not_Rotating,Off,Off,Low,
Off,None
6,Medium,Medium,On,Medium,Medium,On,Medium,Off,In_Range,Off,Rotating,Off,Off,Low
,Off,None
7,Medium,Medium,On,Medium,Medium,On,Medium,Off,In_Range,Off,Rotating,Off,Off,Low
,Off,None
8,Medium,Medium,On,Medium,Medium,On,Medium,Off,In_Range,Off,Rotating,Off,Off,Low
,Off,None
9,Medium,Medium,On,Medium,Medium,On,Medium,Off,In_Range,Off,Rotating,Off,Off,Low
,Off,None
10,Medium,Medium,On,Medium,Medium,On,Medium,Off,In_Range,Off,Rotating,Off,Off,Lo
w,Off,None
```

Figure 7 CSV format, it introduces a comma as separator for each column of MS Excel

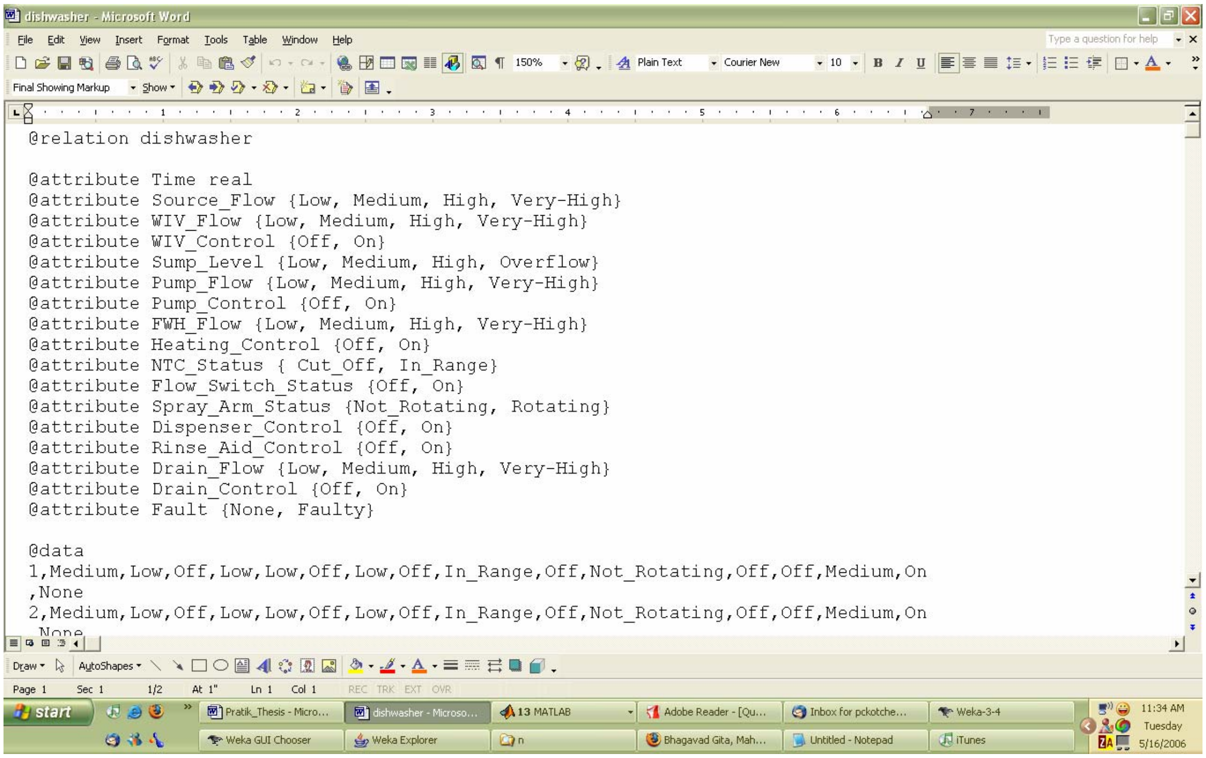


Figure 8 Converting CSV to ARFF and saving as filename.ARFF with "Save Type As" selected as "Plain Text"

c) The data-set in ARFF format is ready for generating rules in WEKA for different machine learning algorithms. After selecting the file into WEKA we need to perform the following steps to generate the rules.

   i. The first step in data mining is to pre-process the data-set to filter out any irregularities in the data set. To be more precise, pre-processing helps in visualizing the data-set results in the form similar to the results obtained in Matlab/Simulink, thus serving as a valid proof for the extraction of data from the qualitative model of the dishwasher. Thus preparing to provide the data to the algorithms as designed in the qualitative model.

ii. The next step is to generate the rules for the Rule Base. This can be performed by applying different algorithms of classify and associate. The classification algorithms have a set of rule based algorithms that can be applied to generate the rules. The disadvantage of using classification algorithms is that, they generate one rule for each class, so if we select each attribute as class and apply the classification algorithm we will get as many rules as the number of attributes. On the other hand, association algorithms generate rules which are sometimes not even seen by the experts and the need for selecting different class is not important in associate algorithms. The disadvantage of associate algorithms is that they tend to generate many redundant rules. The result of generating rules from both classification and association algorithms are shown subsequently in the results section. Among all the algorithms the association algorithm of Tertius seems to generate desirable rules.

The rule base can be further used to develop a questionnaire for the end user with superficial knowledge of the system. Thus, the rule base acts as the knowledge of the expert about the dishwasher system, which can be used by the end user to detect the fault in the dishwasher system, if any. To help the end user to find the solution to problems, he/she needs to provide some factual information he/she observes in the faulty system. The Questionnaire performs the analysis or reasoning of the factual information provided by the end user with the help of the rule base, thus providing solutions to the problems. This is the end product of the Expert System and has not been implemented here. The results shown are of knowledge representation and they inlcude implemention of a qualitative model, extraction of knowledge, and generation of rules of the system.

4.2 Results & Analysis

The experimental results are shown for the methods explained above i.e. Qualitative Model of the dishwasher, Knowledge Extraction and Rule or Knowledge Base.

4.2.1 Qualitative Model Results

The first part of the Qualitative Model, the back processing Matlab code for dishwasher is compiled, which launches the Simulink model of the dishwasher as shown in Figure 2. This simulation when executed initially, it asks for the name of the file where the extracted data needs to be stored. After entering the filename (with extension .xls) and pressing return key, it generates the following qualitative plots of the corresponding dishwasher components. It also shows a display which denotes whether a component is active or not (green => active, red => inactive).

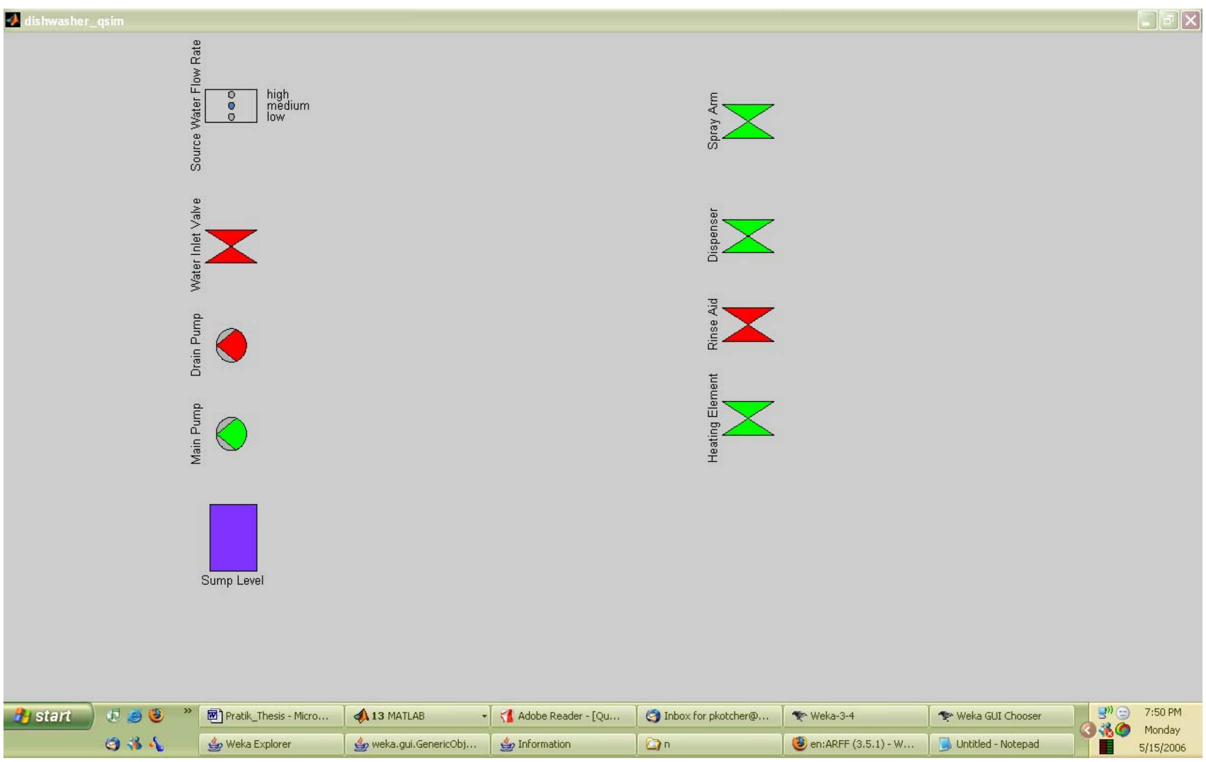


Figure 9 Display showing qualitative status of different components of the dishwasher

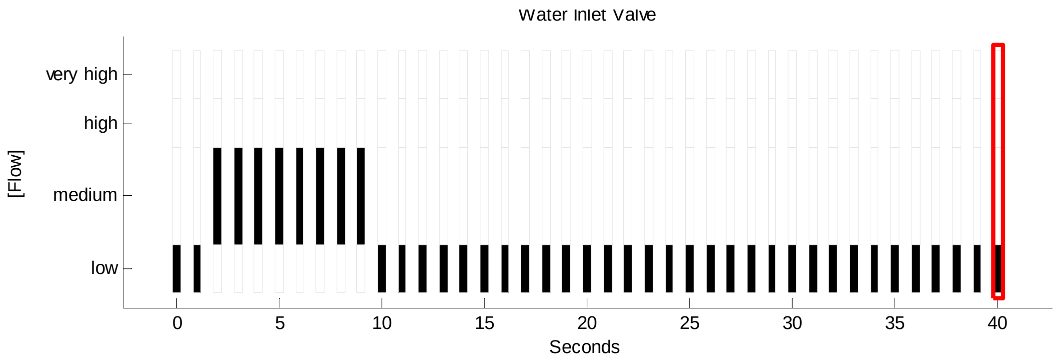


Figure 10 Qualitative Plot of Water Inlet Valve

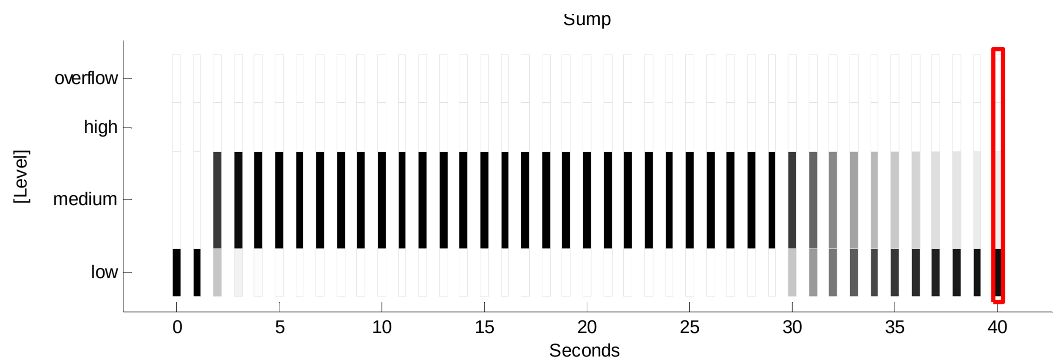


Figure 11 Qualitative Plot of Sump

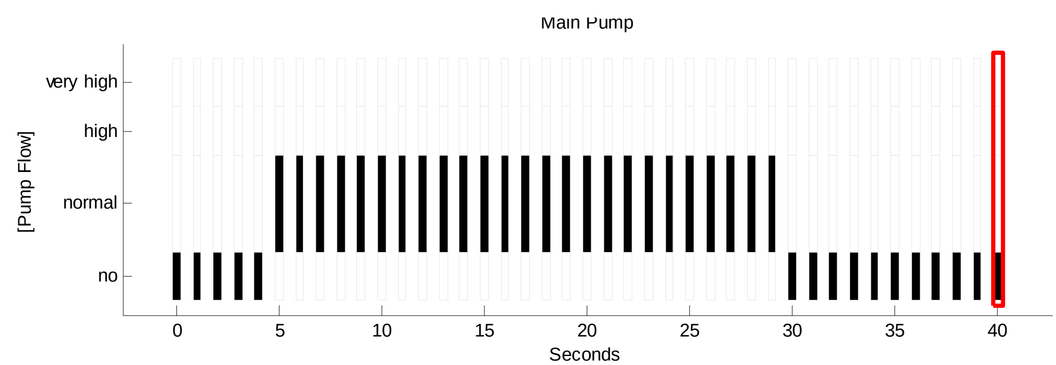


Figure 12 Qualitative Plot of Main Pump

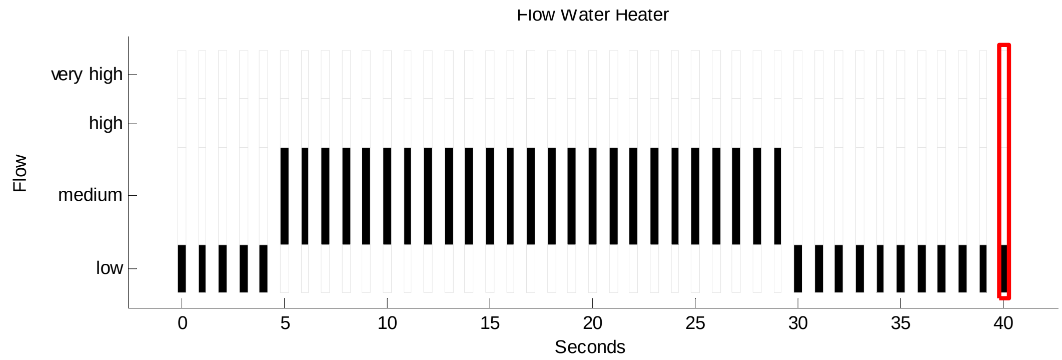


Figure 13 Qualitative Plot of Flow Water Heater

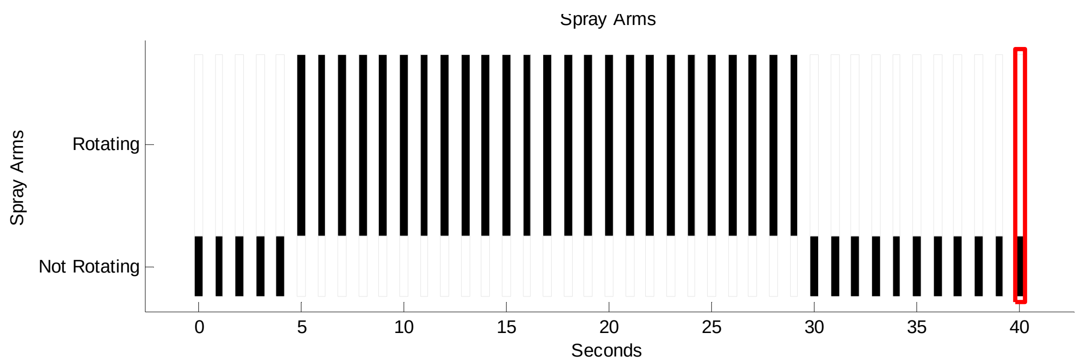


Figure 14 Qualitative Plot of Spray Arm

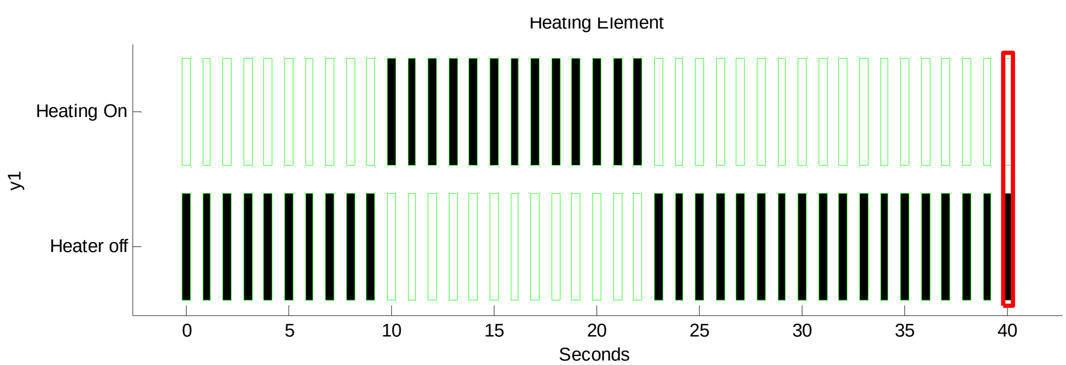


Figure 15 Qualitative Plot of Heating Element

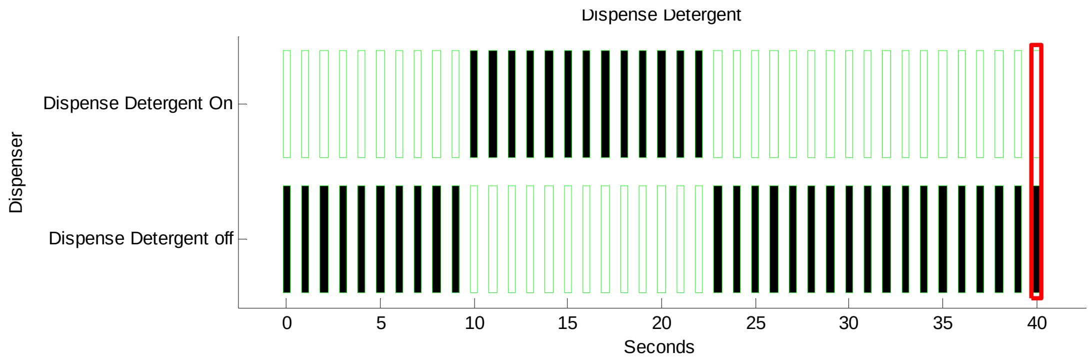


Figure 16 Qualitative Plot of Dispenser Detergent

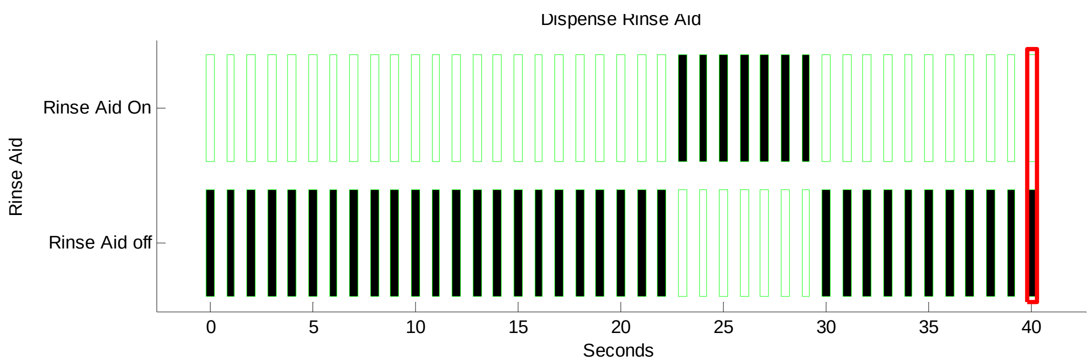


Figure 17 Qualitative Plot of Dispenser Rinse Aid

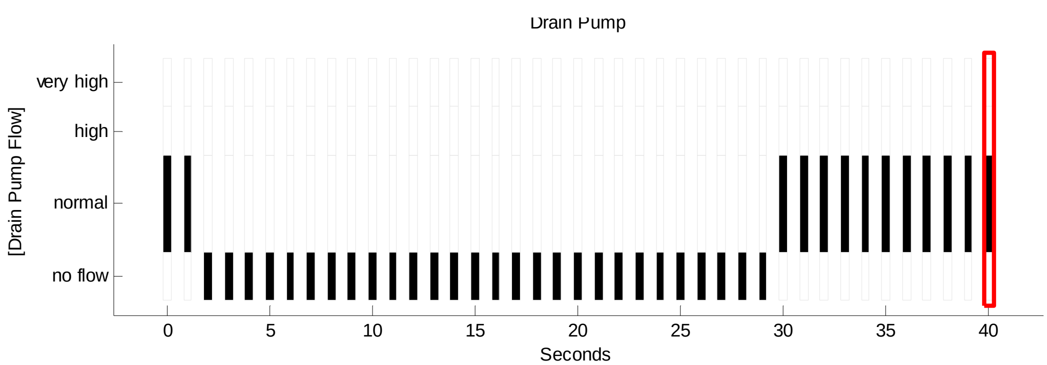
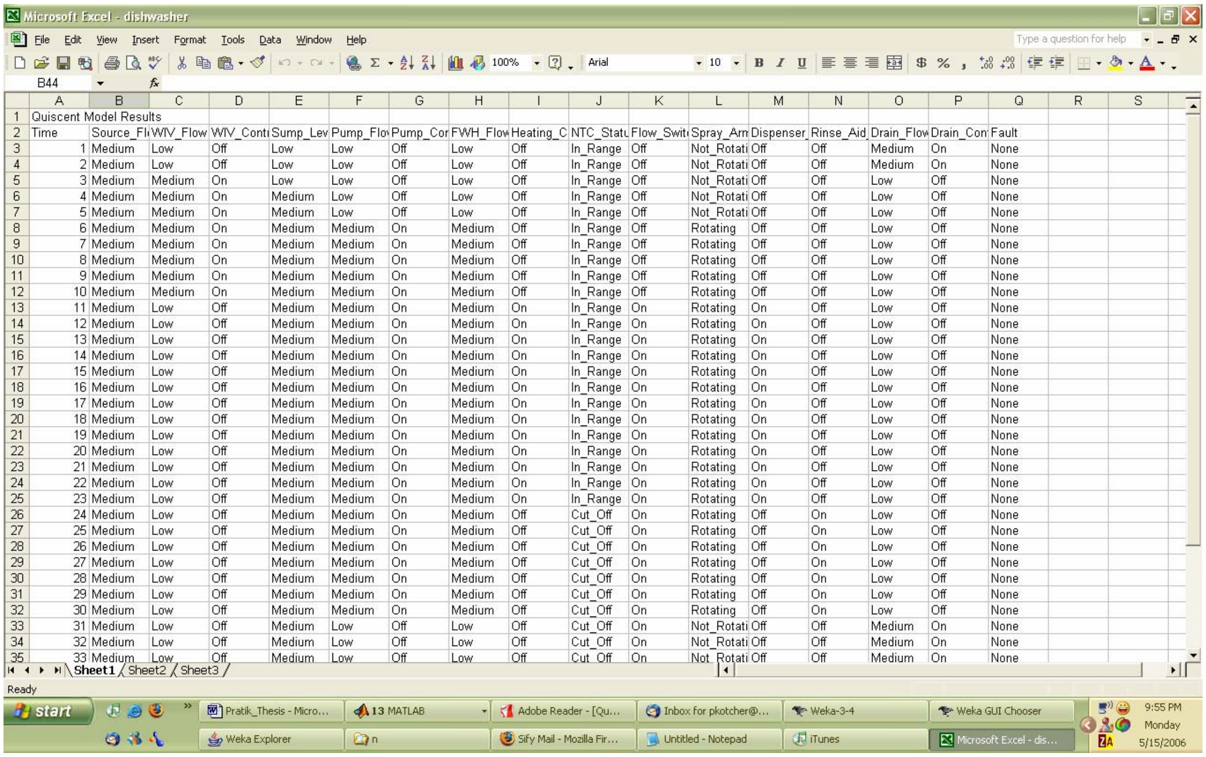


Figure 18 Qualitative Plot of Drain Pump

Figure 10 to Figure 18 shows the qualitative output plots of the dishwasher components. Each plot has 40 instances and the behavior of each component is explained in the section 3.3.2 (Implementation of CSP Algorithm in QuaMo Toolbox).

4.2.2 Knowledge Extraction Results

During the simulation of dishwasher qualitative model in simulink, the qualitative outputs of each block are stored in variables in the Matlab/Simulink workspace along with plotting them on the qualitative plots. After the simulation is completed the values in these variables are stored into a spreadsheet as shown in Figure 14. The storing process is performed by the function called save_data () which consists of enumerating the qualitative values to symbolic values. For the extracted attributes from the qualitative dishwasher model, these symbolic values are stored in the spreadsheet. The function save_data () uses the functions xlswrite8 () and xlsappend8 () to write the extracted knowledge to the spreadsheet from the Matlab/Simulink workspace. For more information on these functions see Appendix II.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Quiscent Model Results																
2	Time	Source_Fl	WIV_Flow	WIV_Cont	Sump_Lev	Pump_Flo	Pump_Cor	FWH_Flow	Heating_C	NTC_Statu	Flow_Swit	Spray_Arm	Dispenser_	Rinse_Aid	Drain_Flow	Drain_Con	Fault
3	1	Medium	Low	Off	Low	Low	Off	Low	Off	In_Range	Off	Not_Rotati	Off	Off	Medium	On	None
4	2	Medium	Low	Off	Low	Low	Off	Low	Off	In_Range	Off	Not_Rotati	Off	Off	Medium	On	None
5	3	Medium	Medium	On	Low	Low	Off	Low	Off	In_Range	Off	Not_Rotati	Off	Off	Low	Off	None
6	4	Medium	Medium	On	Medium	Low	Off	Low	Off	In_Range	Off	Not_Rotati	Off	Off	Low	Off	None
7	5	Medium	Medium	On	Medium	Low	Off	Low	Off	In_Range	Off	Not_Rotati	Off	Off	Low	Off	None
8	6	Medium	Medium	On	Medium	Medium	On	Medium	Off	In_Range	Off	Rotating	Off	Off	Low	Off	None
9	7	Medium	Medium	On	Medium	Medium	On	Medium	Off	In_Range	Off	Rotating	Off	Off	Low	Off	None
10	8	Medium	Medium	On	Medium	Medium	On	Medium	Off	In_Range	Off	Rotating	Off	Off	Low	Off	None
11	9	Medium	Medium	On	Medium	Medium	On	Medium	Off	In_Range	Off	Rotating	Off	Off	Low	Off	None
12	10	Medium	Medium	On	Medium	Medium	On	Medium	Off	In_Range	Off	Rotating	Off	Off	Low	Off	None
13	11	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
14	12	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
15	13	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
16	14	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
17	15	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
18	16	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
19	17	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
20	18	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
21	19	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
22	20	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
23	21	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
24	22	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
25	23	Medium	Low	Off	Medium	Medium	On	Medium	On	In_Range	On	Rotating	On	Off	Low	Off	None
26	24	Medium	Low	Off	Medium	Medium	On	Medium	Off	Cut_Off	On	Rotating	Off	On	Low	Off	None
27	25	Medium	Low	Off	Medium	Medium	On	Medium	Off	Cut_Off	On	Rotating	Off	On	Low	Off	None
28	26	Medium	Low	Off	Medium	Medium	On	Medium	Off	Cut_Off	On	Rotating	Off	On	Low	Off	None
29	27	Medium	Low	Off	Medium	Medium	On	Medium	Off	Cut_Off	On	Rotating	Off	On	Low	Off	None
30	28	Medium	Low	Off	Medium	Medium	On	Medium	Off	Cut_Off	On	Rotating	Off	On	Low	Off	None
31	29	Medium	Low	Off	Medium	Medium	On	Medium	Off	Cut_Off	On	Rotating	Off	On	Low	Off	None
32	30	Medium	Low	Off	Medium	Medium	On	Medium	Off	Cut_Off	On	Rotating	Off	On	Low	Off	None
33	31	Medium	Low	Off	Medium	Low	Off	Low	Off	Cut_Off	On	Not_Rotati	Off	Off	Medium	On	None
34	32	Medium	Low	Off	Medium	Low	Off	Low	Off	Cut_Off	On	Not_Rotati	Off	Off	Medium	On	None
35	33	Medium	Low	Off	Medium	Low	Off	Low	Off	Cut_Off	On	Not_Rotati	Off	Off	Medium	On	None

Figure 19 Extracted Knowledge from the Qualitative Model of Dishwasher System stored in MS Excel

4.2.3 Knowledge or Rule Base Results

The Knowledge or Rule Base is compiled using the extracted knowledge from the qualitative model of the dishwasher. The extracted knowledge is converted into ARFF and provided to WEKA for generating the Knowledge or Rule Base. For a given quiescent or without any faults model, the value of the attribute for all instances of the class @FAULT of the dishwasher dataset is NONE.

The first step of rule generation is to pre-process or filter the data-set. The aim of this filtering process in the experiment here is to prepare the data for WEKA so that the interpretations are the same as in the qualitative model performed in Matlab/Simulink. For this purpose we use the *'weka.filter.unsupervised.attribute.discretize'* algorithm. This algorithm is used to discretize the attribute @TIME over 40 instances.

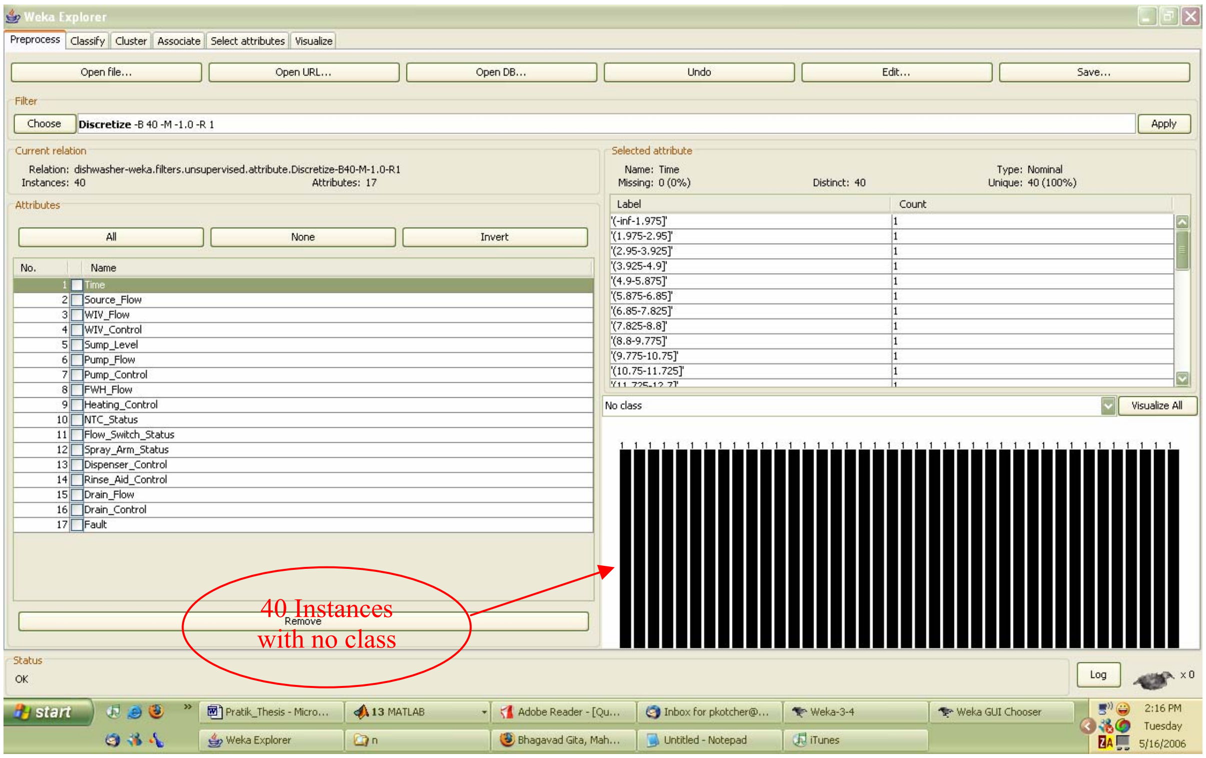


Figure 20 Discretize into 40 instances

The verification is provided by the Visualize Tab in WEKA, here WEKA generates the plots for all attributes vs all attributes as shown in Figure 21. The plots of our interest are for all attributes vs Time as shown in first column of Figure 19 which are similar to the Matlab/Simulink qualitative plots. The class of all the plots is @FAULT with the attribute value NONE shown in blue for all points as this is the dataset of quiescent model of dishwasher.

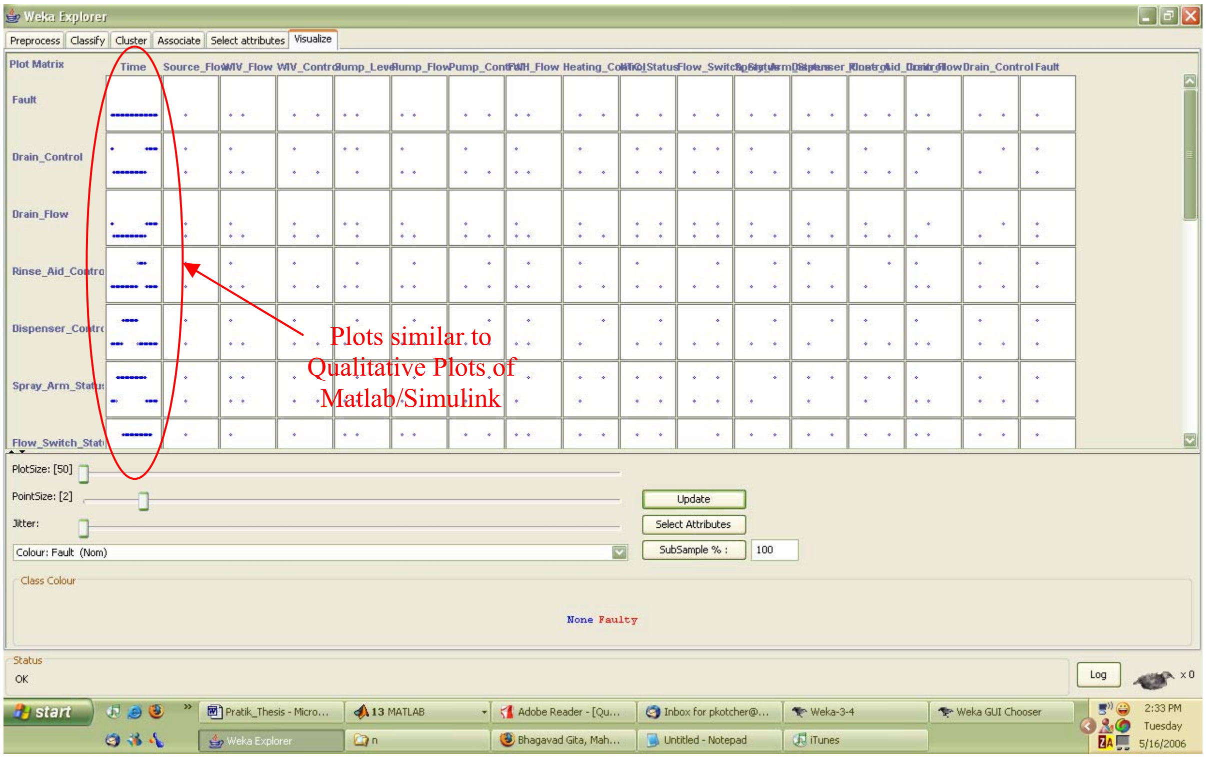


Figure 21 Visualization of the data-set after discretization in WEKA

A larger image of each plot can be obtained by double clicking on the individual plot as shown in Figure 20 (plot of Spray Arm Status vs Time); here we can change the Y-axis to view the plots of other attributes versus Time. Please note that the Blue Crosses are not clearly visible in the plot below.

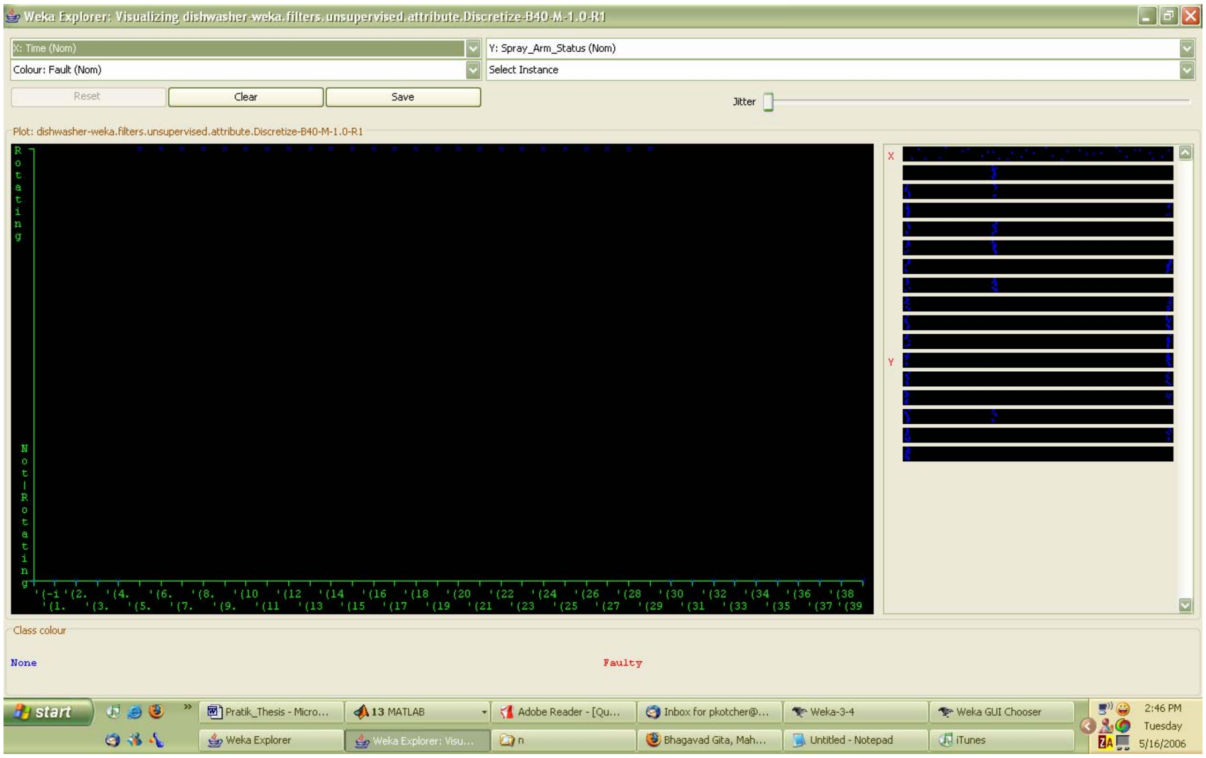


Figure 22 Larger view of Spray Arm Status vs Time

The next step is the investigation of generating the rules for the rule base. As mentioned earlier, different classification and association algorithms were applied to the dishwasher datasets. The results are presented as follows:

The classification algorithms were applied from the "*weka.classifiers.rules*", which provides different algorithms such as RIDOR, PRISM, NNge, JRIP and so on. The classifier algorithms provide rules only for the given class, thus results of PRISM algorithm with each attribute as a class is presented. The reason for choosing PRISM is demonstrated by two experiments. The first experiment was performed in WEKA Experiment Environment for percent correct results generated by the three algorithms, RIDOR, NNge & JRIP. All three algorithms resulted in 100% correct results so they were incomparable and thus inconclusive.

PRISM was not included since it requires only nominal attributes and in the original dataset @Time is a real attribute. After converting the attribute @Time into a nominal value by using the filter discretize we can apply the PRISM algorithm to the dataset. The pre-processing of the dataset was not available during this experimentat; hence it was not included in the first experiment.

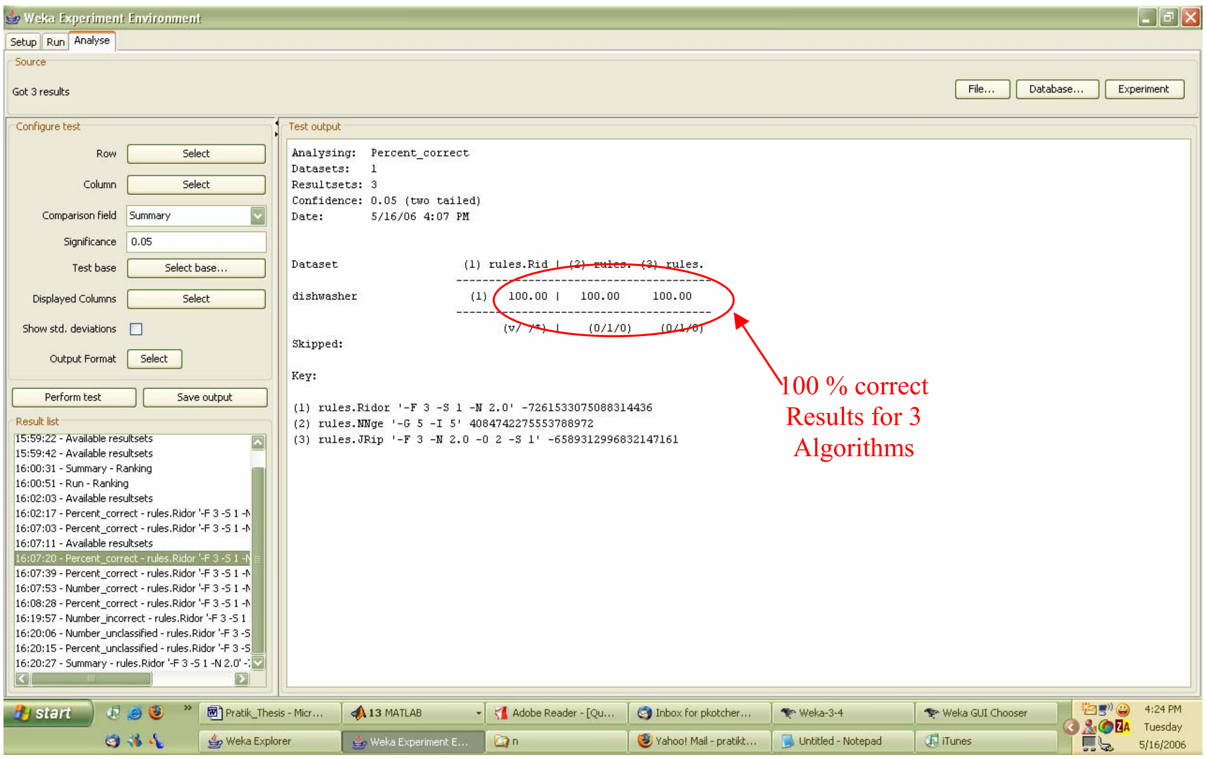


Figure 23 WEKA Experiment Environment with results of RIDOR, NNge & JRIP

The next experiment was to compare the four algorithms against each other by judging the rules generated by each one of them, which are shown in table 3 below, and it appeared that PRISM rules were simplified and close to the rules we needed for the final questionnaire. Thus, the results for PRISM are shown in Figure 24 of WEKA Classifier Tab. Table 4 shows the rules generated in PRISM by selecting each attribute as class.

Table 3 Comparing Classifiers PRISM, RIDOR, NNge, JRIP with Spray_Arm_Status as a Class

Algorithm	Rules Generated
PRISM	If Pump_Flow = Low then Not_Rotating If Pump_Flow = Medium then Rotating
RIDOR	Spray_Arm_Status = Rotating (40.0/15.0) Except (Pump_Flow = Low) => Spray_Arm_Status = Not_Rotating (10.0/0.0) [5.0/0.0]
NNge	class Not_Rotating IF : Time in {'(-inf-1.975]','(1.975-2.95]','(2.95-3.925]','(3.925-4.9]','(4.9-5.875]','(30.25-31.225]','(31.225-32.2]','(32.2-33.175]','(33.175-34.15]','(34.15-35.125]','(35.125-36.1]','(36.1-37.075]','(37.075-38.05]','(38.05-39.025]','(39.025-inf)'} ^ Source_Flow in {Medium} ^ WIV_Flow in {Low,Medium} ^ WIV_Control in {Off,On} ^ Sump_Level in {Low,Medium} ^ Pump_Flow in {Low} ^ Pump_Control in {Off} ^ FWH_Flow in {Low} ^ Heating_Control in {Off} ^ NTC_Status in {Cut_Off,In_Range} ^ Flow_Switch_Status in {Off,On} ^ Dispenser_Control in {Off} ^ Rinse_Aid_Control in {Off} ^ Drain_Flow in {Low,Medium} ^ Drain_Control in {Off,On} ^ Fault in {None} (15) class Rotating IF : Time in {'(5.875-6.85]','(6.85-7.825]','(7.825-8.8]','(8.8-9.775]','(9.775-10.75]','(10.75-11.725]','(11.725-12.7]','(12.7-13.675]','(13.675-14.65]','(14.65-15.625]','(15.625-16.6]','(16.6-17.575]','(17.575-18.55]','(18.55-19.525]','(19.525-20.5]','(20.5-21.475]','(21.475-22.45]','(22.45-23.425]','(23.425-24.4]','(24.4-25.375]','(25.375-26.35]','(26.35-27.325]','(27.325-28.3]','(28.3-29.275]','(29.275-30.25]'} ^ Source_Flow in {Medium} ^ WIV_Flow in {Low,Medium} ^ WIV_Control in {Off,On} ^ Sump_Level in {Medium} ^ Pump_Flow in {Medium} ^ Pump_Control in {On} ^ FWH_Flow in {Medium} ^ Heating_Control in {Off,On} ^ NTC_Status in {Cut_Off,In_Range} ^ Flow_Switch_Status in {Off,On} ^ Dispenser_Control in {Off,On} ^ Rinse_Aid_Control in {Off,On} ^ Drain_Flow in {Low} ^ Drain_Control in {Off} ^ Fault in {None} (25)
JRIP	(Pump_Flow = Low) => Spray_Arm_Status=Not_Rotating (15.0/0.0) => Spray_Arm_Status=Rotating (25.0/0.0)

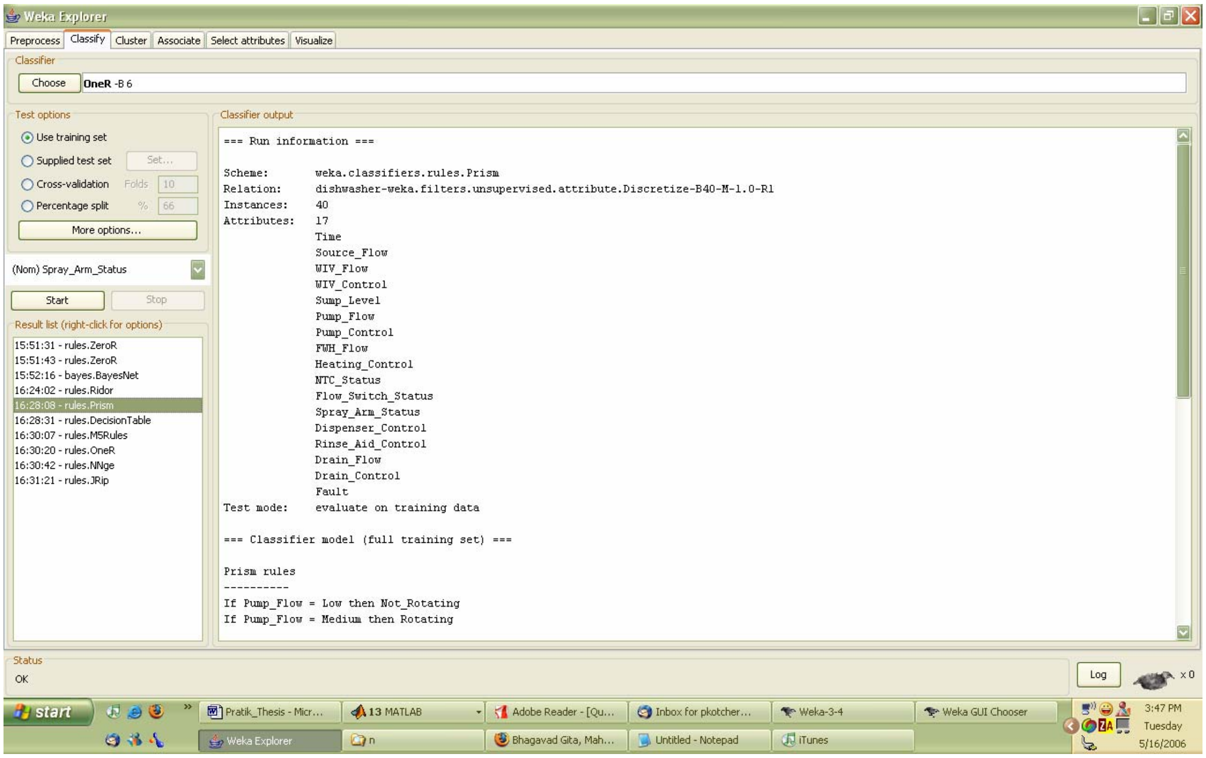


Figure 24 WEKA Classifier Tab

%%%%%%%%%%%%%%% Results of WEKA Classifier PRISM %%%%%%%%%%%%

=== Run information ===

Scheme: weka.classifiers.rules.Prism
Relation: dishwasher-weka.filters.unsupervised.attribute.Discretize-B40-M-1.0-R1
Instances: 40
Attributes: 17
Time
Source_Flow
WIV_Flow
WIV_Control
Sump_Level
Pump_Flow
Pump_Control
FWH_Flow
Heating_Control
NTC_Status
Flow_Switch_Status
Spray_Arm_Status
Dispenser_Control
Rinse_Aid_Control
Drain_Flow
Drain_Control

Fault
Test mode: evaluate on training data

=== Classifier model (full training set) ===

Prism rules

If Pump_Flow = Low then Not_Rotating
If Pump_Flow = Medium then Rotating

Time taken to build model: 0 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	40	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0		
Root mean squared error	0		
Relative absolute error	0 %		
Root relative squared error	0 %		
Total Number of Instances	40		

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	0	1	1	1	Not_Rotating
1	0	1	1	1	Rotating

=== Confusion Matrix ===

```
  a  b   <-- classified as
 15  0 |  a = Not_Rotating
  0 25 |  b = Rotating
```

%%%%%%%%%%%%%%%%%%%%%%%% END %%%%%%%%%%%%%%%%%%%%%%%%

Table 4 PRISM Rules with each Attribute defined as a Class

Attribute	Rule Generated
Time	Generates Rule for each time instant. (too long)
Source_Flow	Medium
WIV_Flow	If WIV_Control = Off then Low If WIV_Control = On then Medium
WIV_Control	If WIV_Flow = Low then Off If WIV_Flow = Medium then On
Sump_Level	If Time = '(-inf-1.975]' then Low If Time = '(1.975-2.95]' then Low If Time = '(31.225-32.2]' then Low If Time = '(32.2-33.175]' then Low If Time = '(33.175-34.15]' then Low If Time = '(34.15-35.125]' then Low If Time = '(35.125-36.1]' then Low If Time = '(36.1-37.075]' then Low If Time = '(37.075-38.05]' then Low If Time = '(38.05-39.025]' then Low If Time = '(39.025-inf)' then Low If Drain_Flow = Low then Medium If Time = '(30.25-31.225]' then Medium
Pump_Flow	If Pump_Control = Off then Low If Pump_Control = On then Medium
Pump_Control	If Pump_Flow = Low then Off If Pump_Flow = Medium then On
FWH_Flow	If Pump_Flow = Low then Low If Pump_Flow = Medium then Medium
Heating_Control	If Dispenser_Control = Off then Off If Dispenser_Control = On then On
NTC_Status	If Rinse_Aid_Control = On then Cut_Off If Time = '(30.25-31.225]' then Cut_Off If Time = '(31.225-32.2]' then Cut_Off If Time = '(32.2-33.175]' then Cut_Off If Time = '(33.175-34.15]' then Cut_Off If Time = '(34.15-35.125]' then Cut_Off If Time = '(35.125-36.1]' then Cut_Off If Time = '(36.1-37.075]' then Cut_Off If Time = '(37.075-38.05]' then Cut_Off If Time = '(38.05-39.025]' then Cut_Off If Time = '(39.025-inf)' then Cut_Off If Heating_Control = On then In_Range If WIV_Flow = Medium then In_Range If Time = '(-inf-1.975]' then In_Range If Time = '(1.975-2.95]' then In_Range

Table 4 (Continued)

Flow_Switch_Status	If WIV_Flow = Medium then Off If Time = '(-inf-1.975]' then Off If Time = '(1.975-2.95]' then Off If Time = '(35.125-36.1]' then Off If Time = '(36.1-37.075]' then Off If Time = '(37.075-38.05]' then Off If Time = '(38.05-39.025]' then Off If Time = '(39.025-inf)' then Off If Heating_Control = On then On If Rinse_Aid_Control = On then On If Time = '(30.25-31.225]' then On If Time = '(31.225-32.2]' then On If Time = '(32.2-33.175]' then On If Time = '(33.175-34.15]' then On If Time = '(34.15-35.125]' then On
Spray_Arm_Status	If Pump_Flow = Low then Not_Rotating If Pump_Flow = Medium then Rotating
Dispenser_Control	If Heating_Control = Off then Off If Heating_Control = On then On
Rinse_Aid_Control	If NTC_Status = In_Range then Off If Pump_Flow = Low then Off If Time = '(23.425-24.4]' then On If Time = '(24.4-25.375]' then On If Time = '(25.375-26.35]' then On If Time = '(26.35-27.325]' then On If Time = '(27.325-28.3]' then On If Time = '(28.3-29.275]' then On If Time = '(29.275-30.25]' then On
Drain_Flow	If Drain_Control = Off then Low If Drain_Control = On then Medium
Drain_Control	If Drain_Flow = Low then Off If Drain_Flow = Medium then On
Fault	None

In the PRISM algorithm, the rules for each attributes were selected as class to generate rules that were specific to the attribute specified as the class. It did not take into account the effects of other attributes.

To generate rules which take into account the complete system we use the algorithms of association. These algorithms give a rule base for the complete system. These algorithms are available under "*weka.associations*". There are three algorithms listed under these association rules, Apriori, PredictiveApriori, and Tertius. PredictiveApriori finds the best 'n' rules maximizing the predictive accuracy, which combines confidence and support in one measure. Tertius has the advantage of finding rules with 'or' connector as well as 'and' connector, whereas APriori finds rules with only 'and' connector. By enabling rocAnalysis in Tertius algorithm we get the confirmation value and frequency of counter-instances. [15] A partial result set of the 3 algorithms with first 5 rules is shown below:

%%%%%%%%%%%% Results from Apriori Association Rules %%%%%%%%%%%%

```
Apriori
=======

Minimum support: 0.65
Minimum metric <confidence>: 0.9
Number of cycles performed: 7

Generated sets of large itemsets:

Size of set of large itemsets L(1): 10

Size of set of large itemsets L(2): 22

Size of set of large itemsets L(3): 19

Size of set of large itemsets L(4): 7

Size of set of large itemsets L(5): 1

Best rules found:

 1. Source_Flow=Medium 40 ==> Fault=None 40    conf:(1)
 2. Fault=None 40 ==> Source_Flow=Medium 40    conf:(1)
 3. Rinse_Aid_Control=Off 33 ==> Source_Flow=Medium Fault=None 33    conf:(1)
 4. Source_Flow=Medium Rinse_Aid_Control=Off 33 ==> Fault=None 33    conf:(1)
 5. Rinse_Aid_Control=Off Fault=None 33 ==> Source_Flow=Medium 33    conf:(1)
```

%%%%%%%%%%%%%%%%%%% END %%%%%%%%%%%%%%%%%%%%

%%%%%%%%% Results from PredictedApriori Association Rules %%%%%%%%%%%

PredictiveApriori
===================
Best rules found:

1. Source_Flow=Medium 40 ==> Fault=None 40   acc:(0.98803)
2. Fault=None 40 ==> Source_Flow=Medium 40   acc:(0.98803)
3. Rinse_Aid_Control=Off 33 ==> Source_Flow=Medium Fault=None 33   acc:(0.98544)
4. WIV_Flow=Low 32 ==> Source_Flow=Medium WIV_Control=Off 32   acc:(0.98496)
5. WIV_Flow=Low 32 ==> Source_Flow=Medium Fault=None 32   acc:(0.98496)

%%%%%%%%%%%%%%%%%%%% END %%%%%%%%%%%%%%%%%%%%%

%%%%%%%%% Results from PredictedApriori Association Rules %%%%%%%%%%%

Tertius
=======

1. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> Pump_Flow = Low or Flow_Switch_Status = Off
2. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> Pump_Flow = Low or Flow_Switch_Status = Off
3. /* 1.000000 1.000000 0.000000 */ Pump_Flow = Medium and Flow_Switch_Status = On ==> Heating_Control = On or Rinse_Aid_Control = On
4. /* 1.000000 1.000000 0.000000 */ Pump_Flow = Medium and Flow_Switch_Status = On ==> Dispenser_Control = On or Rinse_Aid_Control = On
5. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> Pump_Control = Off or Flow_Switch_Status = Off

%%%%%%%%%%%%%%%%%%%% END %%%%%%%%%%%%%%%%%%%%%

For a complete set of rules for the three association algorithms refer the Appendix III.

The above experiments show a framework for developing a Qualitative Model, a framework for extracting the knowledge, and a framework for generating a Rule Base of the dishwasher system from the extracted knowledge. These are the basic components for developing an Expert System for fault diagnosis of dishwasher system. A questionnaire which analyzes the factual information provided by the end user with the aid of the Rule Base is useful in providing solution to the faults in the dishwasher. This completes the Expert System for fault diagnosis of the dishwasher. The idea of developing an Expert System is to provide the end user with the knowledge and analytical ability of an expert in the given system.

An Expert System for fault diagnosis generally deals with modeling the intuitive reasoning ability of an expert. A qualitative model of the system can provide the required information about the system to do fault diagnosis. Also, it is easy to build a qualitative model over a quantitative model. Another advantage of the qualitative model is that it preserves the perspective of the end user, who may not be able to interpret the intricacies of the system from the quantitative model. Moreover, the qualitative model is close to the intuitive sense of the expert and provides solutions in real sense of perception to the end user, without involving the end user into the quantitative details. It is as good as providing the end user with the expert's mind set on the given system. The extraction of knowledge from the system is equivalent to various parameters an expert considers to solve the problem and the development of the rule base requires the analytical capabilities of the expert. Finally, developing a qualitative model as close as possible to the perception of the expert will lay the foundation for developing a foolproof expert based diagnostic system for the end user.

5 Conclusion

An approach to designing a fault diagnosis for a dishwasher system is demonstrated in this research. This approach can be extended to any system in the domain of applications which can be consumer goods, industrial products, chemical processes and so on. An innovative approach was used to design the fault diagnosis using expert system in this research which could prove to be a useful tool for the field technician of Bosch & Siemens Home Appliances Corporation's, Dishwasher Division. The expert system provides the knowledge and capability of the designers/engineers to the technicians while on field which in turn helps reduce costs and downtime of the machine due to improper fault diagnosis by the field technicians.

An innovative approach of applying Expert System in this regard was shown through this research. Traditionally, the field technicians are trained with the working of the dishwasher and provided with the technical manual of the dishwasher with detailed specifications, which are sometimes difficult to comprehend easily and requires the knowledge of the designer to understand. Also, the technician has to refer to the manual from time to time to repair the faulty system, when he is not able to interpret the symptoms he is observing with information given in the manual. The approach given here eliminated the above mentioned difficulties; first, he does not need to comprehend with the technical intricacies of the dishwasher and second, eliminates the need to refer to the technical manual for finding a solution to a problem. The approach given here provided a qualitative view of the dishwasher system as interpreted by the experts of dishwasher. Thus, the technician is provided with the analytical abilities of the expert and the only job he needs to perform while finding faults is to provide with symptoms he is observing in the system. Providing the symptoms to the Expert system or in other words the expert mind, the technician doesn't need look into the intricate details of the system since the expert mind is there with him to help him solve the problem.

The Expert System approach taken here consists of 4 major parts; 1) Designing a Qualitative Model of the dishwasher system, 2) Extracting the Knowledge of the System, 3) Generating a Rule Base of the System, 4) Developing a Questionnaire based on Expert analysis for fault detection or the Expert System Shell. Frameworks for the first three parts were designed and their results are provided in this report. These 3 parts combine to form the knowledge representation in Expert System.

A framework of qualitative approach was used to develop the model of the dishwasher system using the QuaMo Toolbox in Matlab/Simulink. A qualitative model requires approximate information of the system and closely simulates the thinking of an expert's mind of the system. It does not take into account the consideration of the quantitative values and mathematical relationships to model the system. Thus simple symbolic relationships such as LOW, HIGH, NORMAL, ON, OFF and so on are used to describe the system and these symbols are used to provide solutions to the field technicians, thus making it easy for the technician to comprehend information provided by the expert system. The second part was to extract the knowledge of the dishwasher. The important aspect of developing an expert system is that the extracted knowledge must be coincident with the knowledge of the expert, since accurate and detailed knowledge of the dishwasher system will aid in generating accurate and detailed rules for the system. The third part dealt with formatting the extracted knowledge and making it available for WEKA (Machine Learning Tool), which performs the experiment for the generating the Rule Base of the system. Various Classifier & Association algorithms were implemented on the extracted knowledge. Classification algorithms generate one rule for the defined class and hence rules for each attribute defined as a class needs to be generated. The PRISM classifier generated the desired rules set for each attribute, but the rule base generated using this algorithm was quite limited and was not in its entirety. The association rules provided with rules that described the system in its entirety providing with certain rules that were not quite obvious to the experts. The Tertius association algorithm in WEKA seemed to provide quite accurate rules with support and confidence factor for each rule. The fourth and final part was not implemented, but tools of Expert System Shells like Visual Prolog ESTA, e2glite were explored.

Finally, the above approach can be implemented to other domestic systems as well. Thus, providing the field technicians expert help in the field for similar systems.

6 Suggestions for Future Work

Apart from development of the Expert System shell some aspects of the other parts needs to be looked into for improvements. The qualitative model of the dishwasher still needs some additional components like Aqua Sensor, Power Switch, Door Switch were left out due to complexities in design, yet it didn't affect the overall behavior and proving the implementation of the model qualitatively. Adding these components will complete the qualitative model as close to the real model, also the whole model can be simulated for the actual instances or actual value of the time period required for executing the CSP (Customer Service Program). The dataset extracted from the qualitative model seemed to provide quite accurate information when compared with the results of qualitative model as shown by visualizations in WEKA. The generation of Rule base using the association rules, although quite satisfactory but also provided with some redundant rules and the Rule Base needs to be filtered before being provided to the Expert System Shell. A robust Rule Base will result in forming an analytical questionnaire as close possible to the expert sense of reasoning. Finally, to make the decision by the technician more judicious, more datasets related to the failure rates of the components can be added to the existing dataset to generate rules out of some practical approaches taken by the experts apart from the technical aspects of the dishwasher. Also, the cost factor can be included which will make the decision making profitable, for example if a part of a component needs to be replaced and the part doesn't have significant impact on the incurring cost if the whole component is replaced then the whole component can be replaced. This will result in ease of replacement for the technicians. Such knowledge can be gathered and put together to develop an intelligent Expert System.

Additionally, the whole approach taken here can be integrated to create a platform to develop an Expert System for fault diagnosis for any given system. Tools such as the QuaMo Toolbox in Matlab/Simulink for developing qualitative model of any system, MS Excel or any other spreadsheet software for extracted knowledge and storage, WEKA algorithms for generating Rule Base and finally the expert system shell such as that ESTA in Visual Prolog can be integrated together to built a standard platform for developing Qualitative Model based fault diagnostic Expert Systems.

References

[1] Kenneth D Forbus, *CRC Handbook of Computer Science and Engineering*, (CRC Press, 1996), Chap. 32, p.32.1-32.19.

[2] G. Lichtenberg, J. Lunze, J. Schröder, “QuaMo Toolbox for MATLAB/SIMULINK”, Automatic Control Engineering, **45**, 442-443, 1997

[3] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Kewen Yin, and Surya N Kavuri, “A review of process fault detection and diagnosis Part I: Quantitative model-based methods”, Computers in Chemical Engineering **27**, 293-311 (2003a).

[4] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Kewen Yin, and Surya N Kavuri, “A review of process fault detection and diagnosis Part II: Qualitative models and search strategies”, Computers in Chemical Engineering **27**, 313-326 (2003b).

[5] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Kewen Yin, and Surya N Kavuri, “A review of process fault detection and diagnosis Part III: Process History Based Methods”, Computers in Chemical Engineering **27**, 327-346 (2003c).

[6] Kenneth D Forbus, in *Exploring Artificial Intelligence*, edited by Howard E Shrobe (Morgan Kaufmann Publishers, Inc, San Mateo, 1988), Chap. 7, p.239-296.

[7] Randall Davis & Walter Hamscher, in *Exploring Artificial Intelligence*, edited by Howard E Shrobe (Morgan Kaufmann Publishers, Inc, San Mateo, 1988), Chap. 8, p.297-347.

[8] Spyros Tzafestas, in *Expert Systems in Engineering Applications*, edited by Spyros Tzafestas (Springer-Verlag, Berlin, 1993), Chap. 1, p.3-24.

[9] J. Schröder, “QuaMo-Toolbox Reference: Documentation Release 5.2”, Ruhr-Universität Bochum, (2000).

[10] Ian H. Witten & Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques (2nd Ed.),* (Morgan Kaufmann Publishers, Inc, San Mateo, June 2005).

[11] J. Schröder, "QuaMo-Toolbox User's Guide: Documentation Release 5.2", Ruhr-Universität Bochum, (2000).

[12] WekaDoc available online at http://weka.sourceforge.net/wekadoc/index.php/en:Weka_3.5.2

[13] Matlab Documentation available online at http://www.mathworks.com/

[14] J. Schröder, "QuaMo-Toolbox Tutorial: Documentation Release 5.2", Ruhr-Universität Bochum, (2000).

[15] Tertius 1.0 Manual available online at

http://www.cs.bris.ac.uk/Research/MachineLearning/Tertius/

Appendices

I QuaMo Toolbox (Qualitative Modeling Tool in Matlab/Simulink)

Introduction

The qualitative model of the dishwasher system is developed in QuaMo Toolbox. Qualitative Modeling Toolbox (QuaMo Toolbox), a Matlab compatible toolbox used to synthesize, analyze, supervise and control dynamic systems described by qualitative models [2]. The model developed here just harnesses the concept of qualitative modeling and does not use the control and supervision part of QuaMo Toolbox.

Qualitative Models do not refer to numerically precise signals but to symbolic signal values. Engineering knowledge & experience concern to a large extent the qualitative assessment of the system behavior. The QuaMo Toolbox is a flexible platform for research & development in qualitative modeling. The toolbox consists of three parts:

i) Collection of functions mainly used for “generation of automata, space partition & analysis of automata for linear systems”.

ii) QSTOOL module which contains users’ interactive menus for simulation, abstraction of non-linear systems, it also contains methods for analysis of discrete time and discrete event systems.

iii) Collection of Simulink blocks (QLIB & PSLIB), which includes algorithms for applying qualitative model in simulation, prediction observation & diagnosis. It graphically displays the qualitative behavior and to simulate the test algorithm interactively.

Principle of Qualitative Modeling

The QuaMo Toolbox is intended to use with qualitative models, i.e. models which describes the process behavior qualitatively. It is a representation of system behavior as a relation between input-output signals, which are also known as symptoms. Symptoms can be written as logical propositions, thus qualitative models are logical implications.

Proposition of Causes => Proposition of Effects

QuaMo Toolbox is developed for using with the qualitative model in the form of "automata" (Finite State Machines). Thus each state is bounded by a cause-effect relationship called as automata. Logical formulae in automata are associated with relation in the form

P(effects/causes)

P => either Possibility (True/False) or Probability

When it's a possibility, the automata are non-deterministic and for probability automata are stochastic. In general, the automata are dynamic models providing additional information about temporal behavior of the system. The cause-effect relation implies temporal sequence of the state transition as in:

P(effects(k+1) | causes(k))

where, effects(k+1) = {successor state, output}

causes(k) = {current state, input}

This means that no successor state can be generated without the current state and input, i.e. both the current state and input cause the successor state and output, which is the idea behind the State Space Theory and applied for qualitative models.

The following types of models can be developed by QuaMo Toolbox out of which the Stochastic automata are used for the dishwasher system.

i) Nondeterministic automata: These are finite state machines (FSMs) where more than one effect can occur for the same cause as described by cause-effect relation.

ii) Stochastic automata: These are more generalized than nondeterministic automata with probabilistic measure for their cause effect relation.

iii) Semi-Markov processes: These are the stochastic processes similar to stochastic automata but describe the cause-effect relation with additional timing constraint indicating how long the current state of the process can be assumed.

There are two methods to obtain the above models by using QuaMo Toolbox. The first being the identification method which is the modelling method from experiment and the second being the abstraction method. The abstraction for a model is done from the system description, e.g. in the form of a discrete-time state space model

$$x(k+1) = f(x(k),u(k)),\ x(0) = x0$$

$$y(k) = g(x(k),u(k)),$$

where $x(k) \in IR^n$ denotes the state, $u(k) \in IR^m$ the input and $y(k) \in IR^r$ the output.

The underlying idea of the abstraction method is based on the quantized systems. Therefore the quantized system is briefly explained as follow. In many applications the signal values cannot be measured precisely, therefore the structure of the system under consideration can be considered to take the form of "*quantized system*" depicted in Figure 23. Hence, the system described by state space model is controlled through an injector, which transforms the qualitative input value [u(k)] into a real value u(k) at time k. On the other hand, the system output y(k) can only be measured by means of a quantizer which provides a qualitative information [y(k)] about the output at time k.

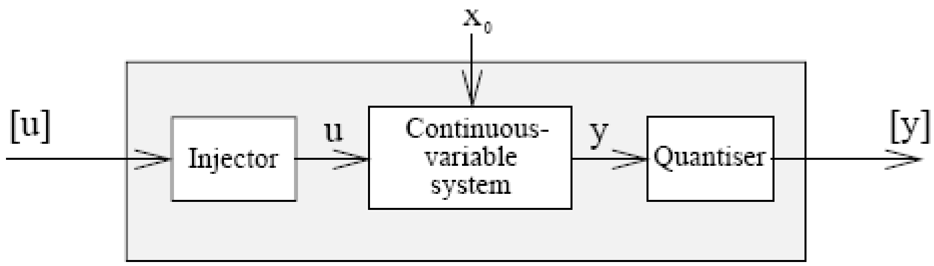


Figure 25 The Quantized System

Mathematically, the injector and the quantizer introduce partitions of the continuous-variable signal spaces IR^m and IR^r. The output quantizer maps the continuous output-space IR^r onto a finite set $N_y = \{0, 1, 2, ..., R\}$ of qualitative values. That is, the quantizer introduces a partition of the output-space IR^r into a finite number of disjoint sets $Q_y(i)$. $Q_y(i)$ denotes the set of outputs $y \in IR^r$ with the same qualitative value i, i.e. [y] = i <=> $y \in Q_y(i)$. Similarly, the input space is partitioned. The abstraction method used in QuaMo Toolbox can help user abstract qualitative models from the quantized system. The main idea to deal with quantized systems is to represent their qualitative behavior by a *stochastic automaton* whose behavioral relation

$$L([x(k+1)], [y(k)] \mid [x(k)], [u(k)])$$

describes the probability that the system, subject to the qualitative input $[u(k)]$ moves from the current qualitative state $[x(k)]$ to the succeeding state $[x(k+1)]$ and simultaneously produces the qualitative output $[y(k)]$.

The QuaMo Toolbox consists of Matlab functions, as used in the back-processing of the dishwasher system model and the Simulink blocks used in the simulation of the dishwasher system model. A list of these functions and blocks used in the qualitative model of the dishwasher is briefly explained here.

Matlab Functions

1] `partition`

Purpose
Constructor for a partition object.
Synopsis

```
Rectangular par = partition(mrf,'r')
partition : par = partition(ticks,'r')
par = partition(lb,ub,'r')
Polygonal par = partition(polys,sinkcell,'p')
partition : par = partition(polygons,'p')
Functional par = partition(flist,ftable,'f')
partition :
Discrete par = partition(dvals,'d')
partition :
Join par = partition(par1,par2,...)
partitions :
(default type is 'r')
```

Description
`partition` generates partition objects based on discrete values, intervals, 2D polygons or 2D functional equations. The data of a partition object is a structure that has varying number of fields according to the type(s) of the partitions it contains. The type information is always stored in the first field of the object as a matrix with two columns and a number of rows that corresponds to the number of dimensions of the partitioned space. `partition` is the constructor function of the partition class.

2] `linss2a`

Purpose
Abstraction of discrete–time qualitative models from discrete-time linear system.
Synopsis

```
L = linss2a(area xs,area us,area ps,AbtsrOpt,
Ad,Bd,Cd,Dd,parx,paru,pary)
```

Description
`linss2a` determines the behavioural relation `L` of the qualitative model of the system given as state space model `Ad,Bd,Cd,Dd` without parameter uncertainties.

$$x(k+1) = A_d \cdot x(k) + B_d \cdot u(k)$$
$$y(k) = C_d \cdot x(k) + D_d \cdot u(k).$$

The parameters `area xs`, `area us`, `area ps` define state, input and parameter space abstraction area, respectively. `area ps` must always be an empty matrix. The abstraction area format is shown in the table below.
The number of rows must match the dimension of the space

First Column :	continuous	positive number, raster points
	discrete	negative number, number of discrete values

Second Column :	continuous	lower limit of abstraction area
	discrete	First discrete value
Third Column :	continuous	upper limit of abstraction area
	discrete	Second discrete value
Fourth Column:	continuous	positive number, raster points
	discrete	Third discrete value

etc. void entries must be set to zero

`AbstrOpt` is a vector with abstraction options and length up to two, where the first element forces the resulting automaton to be stochastic (if set to 1) and the second element forces the program to run silently (if set to 1), i.e. not displaying status messages. Default is `0` for both options. This generated file works only with discrete or rectangular partitions `parx,paru,pary` given as partition object.

`linss2a` calls the compiled file `linss2ac.mex`. The QuaMo–Toolbox comes with precompiled files for WIN-PC and SUN Solaris. On other platforms, you have to compile the C source code `linss2ac.c` before you can use this function.

3] ev2pv

Purpose
Determines the discrete probability distribution of a quantitative vector subject to a given partition of the signal space.
Synopsis
`pv = ev2pv(exact values,par)`
Description
`ev2pv` determines a qualitative vector that consists of probabilities in each dimension of the partition region, to which the given exact value belongs. For polygonal and functional partitions, the probability value is returned at the position of the master parameter of the partition object. At the slaves position a `1` is returned.

4] joinlab

Purpose
Joins labels of different axes for use with `qscope`.
Synopsis
`label = joinlab(label1,label2,...)`
Description
`joinlab` joins labels of different dimensions `label1`, `label2`, etc. The purpose of this function is to add blank lines required for `qscope`.

Simulink Blocks

The QuaMo Toolbox has 2 separate libraries for the simulation block, Qualitative Modeling Library (QLib) & Process Supervision Library (PSLib).

a) QLib Blocks:

The QLib contains algorithms for the applications of qualitative modeling:

• Qualitative simulation and prediction

• Qualitative state observation

• Qualitative control

• Qualitative fault diagnosis and fault tolerant control

Furthermore it contains blocks for quantization, visualization and quantitative simulation.

Concretiser

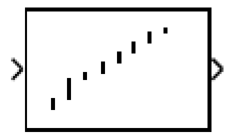


Figure 26 Concretiser

Purpose
Concretiser based on partition objects.
Parameters

Parameter Name	***Value (default)***
Partition	`par`
Information extraction (Popup menu)	**maximum**
	mean/variance
Information generation (Popup menu)	**center/mean**
	random
	interval

Description
The concretiser transforms a qualitative signal given as discrete probability distribution into a continuous-variable signal and is, therefore, the inverse of the quantisation. Obviously, this transformation is not well-defined. Therefore, different methods can be chosen. It is possible to let the continuous variable signal value at each time-point be a random number according to the input distribution. Another possibility is to let the output be the center or mean value of the interval with the highest probability or the distribution, respectively.
Input: Probability vector
Output: Continuous-variable signal vector
The block ”Concretiser” is a part of the QLib based on the S-function `concretiser`.

Probability Generator

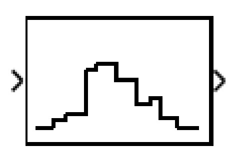


Figure 27 Probability Generator

Purpose
Generates a probability vector from a qualitative number or a qualitative vector.
Parameters

Parameter Name	*Value (default)*
Qualitative dimension or Partition	`par`

Description
Generates a probability distribution from a qualitative vector with respect to a partition `par`, or alternatively from a qualitative number in the qualitative dimension is specified.
Input: Qualitative vector or qualitative number
Output: Probability vector
The block "Probability Generator" is a part of the QLib based on the Sfunction `pvgen`.

Qualitative Scope

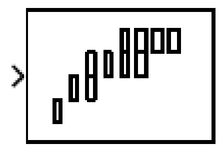


Figure 28 Qualitative Scope

Purpose
Displays probability vectors at discrete time instances.
Parameters

Parameter Name	*Value (default)*
Partition	`par`
Sampling time	`1`
Time steps to display	`20`
Time unit (Popup menu)	**Seconds**, Minutes, Hours
Display unit (Popup menu)	**Seconds**, Minutes, Hours Units, Off
Mark current time instant	Checkbox (**checked**)
Views	`[1 0]`
Variable labels	`joinvl('[y 1]','[y 2]')`
Linguistic labels	`joinlab('','')`

Description
This block displays discrete probability distributions over partitions of signal spaces. 1D and 2D (phase portrait) views are possible. The dimensions to be displayed are determined by the parameter *views*, e.g. `[1 0;2 3]` for a 1D view of partition dimension 1 and a 2D view of partition dimensions 2 and 3. The parameter *Time steps to display* specifies the width of the scope. The parameter *Variable labels* labels the variable axes. The parameter *Linguistic labels* labels the regions of the partition. These parameters are string matrices with strings rowwise. The number of strings has to be identical with the number of qualitative values of all variables. `joinvl` can be used to join variable labels and `joinlab` to join sets of variables labels of the different dimensions, e.g.

```
joinlab(joinvl('high','middle','low')
joinvl('cold','hot'))
```

Input: Probability vector
The block "Qualitative Scope" is a part of the QLib based on the S-function `qscope`.

Qualitative Simulator

Figure 29 Qualitative Simulator

Purpose
Qualitative simulator of a stochastic automaton.
Parameters

Parameter Name	*Value (default)*
Behaviour matrix	`L`
Initial state distribution	`pz0`
Sampling time	`Ts`
Cutoff–probability	`0.0`
Normalise state distribution	Checkbox (**unchecked**)
Normalise output distribution	Checkbox (**unchecked**)
Automatically start with initial state distribution	Checkbox (**checked**)
Output (Popup menu)	**output distribution** state distribution output and state distribution randomised output (qual. number) randomised state (qual. number) randomised state and output

Description
By means of this block the behaviour of the stochastic automaton described by the behavioural matrix `L` can be simulated. There are two possible ways of simulation: On one hand, the stochastic automaton can be simulated like a stochastic process being in a unique state and giving a unique output according to the probabilities given by `L`. On the other hand, the state and the output can be probability distributions. The *Cutoff-probability* parameter gives the bound for neglection of less probable behaviours. The default value 0.0 shows all possible behaviours of the stochastic automaton.
Input: Input distribution (probability vector)
Output: Qualitative state and/or output or probability vector or qualitative number according to chosen output format
The block ”Qualitative Simulator” is a part of the QLib based on the Sfunction `quasi`.

`Quantiser`

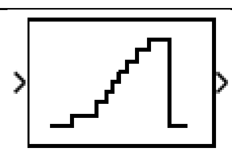


Figure 30 Quantizer

Purpose
Signal quantizer with respect to a partition object.

Parameters

Parameter Name	*Value (default)*
Partition	`par`
Output (Popup menu)	**qualitative number** qualitative vector probability distribution

Description

This block is used to transform a quantitative signal into a qualitative signal with respect to a partition `par`.

Input: Continuous-variable vector

Output: Qualitative number, qualitative vector or probability distribution

The block "Quantiser" is a part of the QLib based on the S-function `quantiser`.

b) PSLib Blocks

The PSLib contains blocks for interactive simulation and process variable visualisation. The library offers common process devices like valve, tank, control panel etc.

• An important parameter for most of the PSLib blocks is called Window Control and it gives the user the opportunity to decide whether each component appears in its own window or all the components in the model appear in one window.

• Another parameter used in most blocks is Signal Name which gives the user the opportunity to label each input or output signal with an individual name.

Qualitative Input

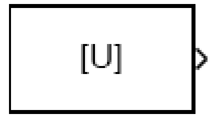


Figure 31 Qualitative Input

Purpose
Creates an interactive qualitative input button array.

Parameters

Parameter Name	*Value (default)*
Number/Names of Buttons	`2`
Default Button Number	`1`
Sampling time	`Ts`
Window Control (Popup menu)	**Own Control Window** Subsystem Model Window
Signal Name	`'Signal name'`
Output (Popup menu)	**Qualitative number** Probability distribution

Description
This block generates an interactive control panel for different qualitative inputs. The parameter *Number/Names of Buttons* gives the user the opportunity to decide between two different ways of labeling:

• By number, the value `n` creates n buttons labelled from 1 to n
• By name, with the entry e.g. `str2mat('one','two','three')`, the parameter assignment creates three buttons labelled with `'one'`, `'two'` and `'three'`.

Output: Qualitative signal as qualitative number or probability distribution
The block “Qualitative Input” is a part of the PSLib based on the S-function `gqin`.

Tank

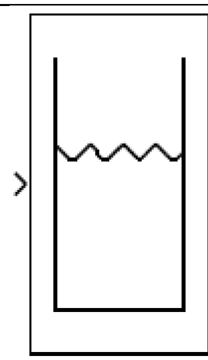


Figure 32 Tank

Purpose
Creates a graphical display of a tank and its liquid level.

Parameters

Parameter Name	*Value (default)*
Minimum level	`0`
Maximum level	`1`
Sampling time	`Ts`

Signal Name 'Signal name'

Description

This block graphically displays a tank and its liquid level according to the input signal.

Input: Continuous-variable scalar signal

The block "Tank" is a part of the PSLib based on the S-function `gtank`.

`Valve`

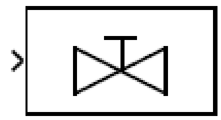


Figure 33 Valve

Purpose

Draws a valve symbol.

Parameters

Parameter Name	***Value (default)***
Sampling time	`Ts`
Signal Name	`'Signal name'`

Description

Draws a valve where the color depends on the input signal. Red color means that the input signal is zero. Otherwise the color of the valve is green.

Input: Variable scalar signal

The block "Valve" is a part of the PSLib based on the S-function `gvalve`.

The above QuaMo Toolbox functions and blocks are available in the QuaMo Toolbox Reference Manual. [9]

II MS Excel write functions from Matlab

Special Functions were obtained from www.mathworks.com, Matlab's home website at the File Exchange community for writing into MS Excel. The functions, xlswrite8() & xlsappend8() were useful to save the extracted knowledge from the Qualitative Model of dishwasher into MS Excel format. These functions are explained below:

`xlswrite8()`

Purpose

Write Excel spreadsheets natively (v97-2003).

Examples

XLSWRITE8(M, 'FILENAME') creates a new spreadsheet FILENAME and writes the matrix M into it. M can be of type double, int32, char, cell or struct. For structs the fieldnames are taken as column headers.

XLSWRITE8(M, 'FILENAME', R, C) writes starting at an offset of R rows and an offset of C columns. R and C are zero-based, so R = C = 0 specifies the top left cell in the spreadsheet.

XLSWRITE8(M, 'FILENAME', R, C, HEAD, COL) adds a header and a column header. HEAD must be a char array or a cell array of strings (for multiple lines). COL is always a cell array of strings.

It's not possible to write into a distinct sheet directly. But you can use this workaround: create a file with xlswrite8 and fill it with xlsappend8.

`xlsappend8()`

Purpose
Append data to an existing Excel spreadsheet (natively).
Examples
XLSAPPEND8(M, 'FILENAME', SHEET) writes matrix M into the selected SHEET in an existing spreadsheet FILENAME. M can be of type double, int32, char, cell or struct. For structs the fieldnames are taken as column headers. SHEET indicates the sheet in which the data will be written, possible are a name or an index (1 based). If SHEET is omitted the first sheet is used.

XLSAPPEND8(M, 'FILENAME', SHEET, R, C) writes starting at an offset of R rows and C columns. R and C are zero-based, so R = C = 0 specifies the top left cell in the spreadsheet.

XLSAPPEND8(M, 'FILENAME', SHEET, R, C, HEAD, COL, SHEET) adds a header and a column header. HEAD must be a char array or a cell array of strings (for multiple lines). COL is always a cell array of strings.

III WEKA (Machine Learning Tool)

The Waikato Environment for Knowledge Analysis (WEKA) [10] is a comprehensive suite of Java class libraries that implement many state-of-the-art machine learning and data mining algorithms. WEKA is used to generate the Rule Base for the extracted qualitative knowledge of the dishwasher system. WEKA provides implementation of learning algorithms that you can easily apply to your dataset. It also includes variety of tools for transforming datasets, one can pre-process a dataset, feed it into a learning scheme and analyze the resulting classifier and its performance.

The workbench includes methods for all standard data mining problems: regression, classification, clustering, association rule mining and attribute selection. Getting to know the data is an integral part of the work and many data visualization facilities and data pre-processing tools are provided. All algorithms take their input in the form of a single relational table in the ARFF (Attribute Relation File Format) format.

The easiest way to use WEKA is through a graphical user interface called the Explorer. This gives access to all of its facilities using menu selection and form filling. There are two other graphical user interfaces to WEKA. The Knowledge Flow interface allows one to design configurations for streamed data processing. The Experimenter, is designed to help one answer a basic practical question when applying classification and regression techniques: which methods and parameter values work best for the given problem? The Explorer and Experimenter were used in the research here to generate the Rule Base and analyze the results from different classifier algorithms respectively.

ARFF Format

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files have two distinct sections. The first section is the Header information, which is followed the Data information. The Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types. The ARFF Data section of the file contains the data declaration line and the actual instance lines.

The @relation Declaration

The relation name is defined as the first line in the ARFF file. The format is:

@relation <relation-name>

where *<relation-name>* is a string. The string must be quoted if the name includes spaces.

The @attribute Declarations

Attribute declarations take the form of an ordered sequence of **@attribute** statements. Each attribute in the data set has its own **@attribute** statement which uniquely defines the name of that attribute and its data type. The order the attributes are declared indicates the column position in the data section of the file. For example, if an attribute is the third one declared then WEKA expects that all that attributes values will be found in the third comma delimited column.

The format for the **@attribute** statement is:

@attribute <attribute-name> <datatype>

where the *<attribute-name>* must start with an alphabetic character. If spaces are to be included in the name then the entire name must be quoted.

The <datatype> can be any of the four types supported by Weka:

numeric

integer is treated as *numeric*

real is treated as *numeric*

<nominal-specification>

string

date [<date-format>]

relational for multi-instance data (for future use)

The keywords **numeric**, **real**, **integer**, **string** and **date** are case insensitive.

The @data Declaration

The **@data** declaration is a single line denoting the start of the data segment in the file. The format is:

@data

Each instance is represented on a single line, with carriage returns denoting the end of the instance. A percent sign (%) introduces a comment, which continues to the end of the line. Attribute values for each instance are delimited by commas (or spaces?). They must appear in the order that they were declared in the header section (i.e. the data corresponding to the nth **@attribute** declaration is always the nth field of the attribute).

The Explorer

A summary of the tabs of Explorer is given below and the algorithms applied:

a) *Preprocess:* Choose the dataset and modify it in various ways.
b) *Classify:* Train learning schemes that perform classification or regression and evaluate them.
c) *Cluster:* Learn clusters for the dataset.
d) *Associate:* Learn association rules for the data and evaluate them.
e) *Select Attributes:* Select the most relevant aspects in the dataset.
f) *Visualize:* View different 2-D plots of the data and interact with them.

The extracted knowledge of the dishwasher is converted into ARFF format and different algorithms of the above types are applied to the dataset are briefed below:

`weka.filters.unsupervised.attribute.Discretize`

SYNOPSIS
An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes. Discretization is by simple binning. Skips the class attribute if set.

OPTIONS
attributeIndices -- Specify range of attributes to act on. This is a comma separated list of attribute indices, with "first" and "last" valid values. Specify an inclusive range with "-". E.g: "first-3,5,6-10,last".

bins -- Number of bins.

desiredWeightOfInstancesPerInterval -- Sets the desired weight of instances per interval for equal-frequency binning.

findNumBins -- Optimize number of equal-width bins using leave-one-out. Doesn't work for equal-frequency binning

invertSelection -- Set attribute selection mode. If false, only selected (numeric) attributes in the range will be discretized; if true, only non-selected attributes will be discretized.

makeBinary -- Make resulting attributes binary.

useEqualFrequency -- If set to true, equal-frequency binning will be used instead of equal-width binning.

`weka.classifiers.rules.JRip`

SYNOPSIS
This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which was proposed by William W. Cohen as an optimized version of IREP.

OPTIONS
checkErrorRate -- Whether check for error rate >= 1/2 is included in stopping criterion.

debug -- Whether debug information is output to the console.

folds -- Determines the amount of data used for pruning. One fold is used for pruning, the rest for growing the rules.

minNo -- The minimum total weight of the instances in a rule.

optimizations -- The number of optimization runs.

seed -- The seed used for randomizing the data.

usePruning -- Whether pruning is performed.

`weka.classifiers.rules.NNge`

SYNOPSIS
Nearest-neighbor-like algorithm using non-nested generalized exemplars (which are hyperrectangles that can be viewed as if-then rules).

OPTIONS
debug -- If set to true, classifier may output additional info to the console.

numAttemptsOfGeneOption -- Sets the number of attempts for generalization.

numFoldersMIOption -- Sets the number of folder for mutual information.

`weka.classifiers.rules.Prism`

SYNOPSIS
Class for building and using a PRISM rule set for classification. Can only deal with nominal attributes. Can't deal with missing values. Doesn't do any pruning.

OPTIONS
debug -- If set to true, classifier may output additional info to the console.

`weka.classifiers.rules.Ridor`

SYNOPSIS
The implementation of a RIpple-DOwn Rule learner. It generates a default rule first and then the exceptions for the default rule with the least (weighted) error rate. Then it generates the "best" exceptions for each exception and iterates until pure. Thus it performs a tree-like expansion of exceptions.The exceptions are a set of rules that predict classes other than the default. IREP is used to generate the exceptions.

OPTIONS
debug -- If set to true, classifier may output additional info to the console.

folds -- Determines the amount of data used for pruning. One fold is used for pruning, the rest for growing the rules.

majorityClass -- Whether the majority class is used as default.

minNo -- The minimum total weight of the instances in a rule.

seed -- The seed used for randomizing the data.

shuffle -- Determines how often the data is shuffled before a rule is chosen. If > 1, a rule is learned multiple times and the most accurate rule is chosen.

wholeDataErr -- Whether worth of rule is computed based on all the data or just based on data covered by rule.

`weka.associations.Apriori`

SYNOPSIS
Finds association rules.

OPTIONS
delta -- Iteratively decrease support by this factor. Reduces support until min support is reached or required number of rules has been generated.

lowerBoundMinSupport -- Lower bound for minimum support.

metricType -- Set the type of metric by which to rank rules. Confidence is the proportion of the examples covered by the premise that are also covered by the consequence. Lift is confidence divided by the proportion of all examples that are covered by the consequence. This is a measure of the importance of the association that is independent of support. Leverage is the proportion of additional examples covered by both the premise and consequence above those expected if the premise and consequence were independent of each other. The total number of examples that this represents is presented in brackets following the leverage. Conviction is another measure of departure from independence and furthermore takes into account implicaton. Conviction is given by P(premise)P(!consequence) / P(premise, !consequence).

minMetric -- Minimum metric score. Consider only rules with scores higher than this value.

numRules -- Number of rules to find.

removeAllMissingCols -- Remove columns with all missing values.

significanceLevel -- Significance level. Significance test (confidence metric only).

upperBoundMinSupport -- Upper bound for minimum support. Start iteratively decreasing minimum support from this value.

`weka.associations.PredictiveApriori`

SYNOPSIS
Finds association rules sorted by predictive accuracy.

OPTIONS
numRules -- Number of rules to find.

`weka.associations.Tertius`

SYNOPSIS
Finds rules according to confirmation measure.

OPTIONS
classIndex -- Index of the class attribute. If set to 0, the class will be the last attribute.

classification -- Find only rules with the class in the head.

confirmationThreshold -- Minimum confirmation of the rules.

confirmationValues -- Number of best confirmation values to find.

frequencyThreshold -- Minimum proportion of instances satisfying head and body of rules

hornClauses -- Find rules with a single conclusion literal only.

missingValues -- Set the way to handle missing values. Missing values can be set to match any value, or never match values or to be significant and possibly appear in rules.

negation -- Set the type of negation allowed in the rule. Negation can be allowed in the body, in the head, in both or in none.

noiseThreshold -- Maximum proportion of counter-instances of rules. If set to 0, only satisfied rules will be given.

numberLiterals -- Maximum number of literals in a rule.

repeatLiterals -- Repeated attributes allowed.

rocAnalysis -- Return TP-rate and FP-rate for each rule found.

valuesOutput -- Give visual feedback during the search. The current best and worst values can be output either to stdout or to a separate window.

The Experimenter

The Weka Experiment Environment enables the user to create, run, modify, and analyse experiments in a more convenient manner than is possible when processing the schemes individually. For example, the user can create an experiment that runs several schemes against a series of datasets and then analyse the results to determine if one of the schemes is (statistically) better than the other schemes. For more on the Experimenter refer WekaDoc [give reference]

IV Results of Association Rules

The results of the 3 different Association algorithms are shown below:

1] APRIORI

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 100 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0
Relation:     dishwasher-weka.filters.unsupervised.attribute.Discretize-B40-M-1.0-R1
Instances:    40
Attributes:   17
              Time
              Source_Flow
              WIV_Flow
              WIV_Control
              Sump_Level
              Pump_Flow
              Pump_Control
              FWH_Flow
              Heating_Control
              NTC_Status
              Flow_Switch_Status
              Spray_Arm_Status
              Dispenser_Control
              Rinse_Aid_Control
              Drain_Flow
```

Drain_Control
Fault
=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.65
Minimum metric <confidence>: 0.9
Number of cycles performed: 7

Generated sets of large itemsets:

Size of set of large itemsets L(1): 10

Size of set of large itemsets L(2): 22

Size of set of large itemsets L(3): 19

Size of set of large itemsets L(4): 7

Size of set of large itemsets L(5): 1

Best rules found:

1. Source_Flow=Medium 40 ==> Fault=None 40 conf:(1)
2. Fault=None 40 ==> Source_Flow=Medium 40 conf:(1)
3. Rinse_Aid_Control=Off 33 ==> Source_Flow=Medium Fault=None 33 conf:(1)
4. Source_Flow=Medium Rinse_Aid_Control=Off 33 ==> Fault=None 33 conf:(1)
5. Rinse_Aid_Control=Off Fault=None 33 ==> Source_Flow=Medium 33 conf:(1)
6. Rinse_Aid_Control=Off 33 ==> Fault=None 33 conf:(1)
7. Rinse_Aid_Control=Off 33 ==> Source_Flow=Medium 33 conf:(1)
8. WIV_Flow=Low 32 ==> Source_Flow=Medium WIV_Control=Off Fault=None 32 conf:(1)
9. WIV_Control=Off 32 ==> Source_Flow=Medium WIV_Flow=Low Fault=None 32 conf:(1)
10. Source_Flow=Medium WIV_Flow=Low 32 ==> WIV_Control=Off Fault=None 32 conf:(1)
11. Source_Flow=Medium WIV_Control=Off 32 ==> WIV_Flow=Low Fault=None 32 conf:(1)
12. WIV_Flow=Low WIV_Control=Off 32 ==> Source_Flow=Medium Fault=None 32 conf:(1)
13. WIV_Flow=Low Fault=None 32 ==> Source_Flow=Medium WIV_Control=Off 32 conf:(1)
14. WIV_Control=Off Fault=None 32 ==> Source_Flow=Medium WIV_Flow=Low 32 conf:(1)
15. Source_Flow=Medium WIV_Flow=Low WIV_Control=Off 32 ==> Fault=None 32 conf:(1)
16. Source_Flow=Medium WIV_Flow=Low Fault=None 32 ==> WIV_Control=Off 32 conf:(1)
17. Source_Flow=Medium WIV_Control=Off Fault=None 32 ==> WIV_Flow=Low 32 conf:(1)
18. WIV_Flow=Low WIV_Control=Off Fault=None 32 ==> Source_Flow=Medium 32 conf:(1)
19. WIV_Flow=Low 32 ==> WIV_Control=Off Fault=None 32 conf:(1)
20. WIV_Control=Off 32 ==> WIV_Flow=Low Fault=None 32 conf:(1)
21. WIV_Flow=Low WIV_Control=Off 32 ==> Fault=None 32 conf:(1)
22. WIV_Flow=Low Fault=None 32 ==> WIV_Control=Off 32 conf:(1)
23. WIV_Control=Off Fault=None 32 ==> WIV_Flow=Low 32 conf:(1)
24. Sump_Level=Medium 32 ==> Source_Flow=Medium Fault=None 32 conf:(1)
25. Source_Flow=Medium Sump_Level=Medium 32 ==> Fault=None 32 conf:(1)
26. Sump_Level=Medium Fault=None 32 ==> Source_Flow=Medium 32 conf:(1)
27. WIV_Control=Off 32 ==> Source_Flow=Medium Fault=None 32 conf:(1)
28. Source_Flow=Medium WIV_Control=Off 32 ==> Fault=None 32 conf:(1)
29. WIV_Control=Off Fault=None 32 ==> Source_Flow=Medium 32 conf:(1)
30. WIV_Flow=Low 32 ==> Source_Flow=Medium Fault=None 32 conf:(1)

31. Source_Flow=Medium WIV_Flow=Low 32 ==> Fault=None 32   conf:(1)
32. WIV_Flow=Low Fault=None 32 ==> Source_Flow=Medium 32   conf:(1)
33. WIV_Flow=Low 32 ==> Source_Flow=Medium WIV_Control=Off 32   conf:(1)
34. WIV_Control=Off 32 ==> Source_Flow=Medium WIV_Flow=Low 32   conf:(1)
35. Source_Flow=Medium WIV_Flow=Low 32 ==> WIV_Control=Off 32   conf:(1)
36. Source_Flow=Medium WIV_Control=Off 32 ==> WIV_Flow=Low 32   conf:(1)
37. WIV_Flow=Low WIV_Control=Off 32 ==> Source_Flow=Medium 32   conf:(1)
38. Sump_Level=Medium 32 ==> Fault=None 32   conf:(1)
39. WIV_Control=Off 32 ==> Fault=None 32   conf:(1)
40. WIV_Flow=Low 32 ==> Fault=None 32   conf:(1)
41. WIV_Flow=Low 32 ==> WIV_Control=Off 32   conf:(1)
42. WIV_Control=Off 32 ==> WIV_Flow=Low 32   conf:(1)
43. Sump_Level=Medium 32 ==> Source_Flow=Medium 32   conf:(1)
44. WIV_Control=Off 32 ==> Source_Flow=Medium 32   conf:(1)
45. WIV_Flow=Low 32 ==> Source_Flow=Medium 32   conf:(1)
46. Drain_Flow=Low 28 ==> Source_Flow=Medium Drain_Control=Off Fault=None 28   conf:(1)
47. Drain_Control=Off 28 ==> Source_Flow=Medium Drain_Flow=Low Fault=None 28   conf:(1)
48. Source_Flow=Medium Drain_Flow=Low 28 ==> Drain_Control=Off Fault=None 28   conf:(1)
49. Source_Flow=Medium Drain_Control=Off 28 ==> Drain_Flow=Low Fault=None 28   conf:(1)
50. Drain_Flow=Low Drain_Control=Off 28 ==> Source_Flow=Medium Fault=None 28   conf:(1)
51. Drain_Flow=Low Fault=None 28 ==> Source_Flow=Medium Drain_Control=Off 28   conf:(1)
52. Drain_Control=Off Fault=None 28 ==> Source_Flow=Medium Drain_Flow=Low 28   conf:(1)
53. Source_Flow=Medium Drain_Flow=Low Drain_Control=Off 28 ==> Fault=None 28   conf:(1)
54. Source_Flow=Medium Drain_Flow=Low Fault=None 28 ==> Drain_Control=Off 28   conf:(1)
55. Source_Flow=Medium Drain_Control=Off Fault=None 28 ==> Drain_Flow=Low 28   conf:(1)
56. Drain_Flow=Low Drain_Control=Off Fault=None 28 ==> Source_Flow=Medium 28   conf:(1)
57. Drain_Flow=Low 28 ==> Drain_Control=Off Fault=None 28   conf:(1)
58. Drain_Control=Off 28 ==> Drain_Flow=Low Fault=None 28   conf:(1)
59. Drain_Flow=Low Drain_Control=Off 28 ==> Fault=None 28   conf:(1)
60. Drain_Flow=Low Fault=None 28 ==> Drain_Control=Off 28   conf:(1)
61. Drain_Control=Off Fault=None 28 ==> Drain_Flow=Low 28   conf:(1)
62. Drain_Control=Off 28 ==> Source_Flow=Medium Fault=None 28   conf:(1)
63. Source_Flow=Medium Drain_Control=Off 28 ==> Fault=None 28   conf:(1)
64. Drain_Control=Off Fault=None 28 ==> Source_Flow=Medium 28   conf:(1)
65. Drain_Flow=Low 28 ==> Source_Flow=Medium Fault=None 28   conf:(1)
66. Source_Flow=Medium Drain_Flow=Low 28 ==> Fault=None 28   conf:(1)
67. Drain_Flow=Low Fault=None 28 ==> Source_Flow=Medium 28   conf:(1)
68. Drain_Flow=Low 28 ==> Source_Flow=Medium Drain_Control=Off 28   conf:(1)
69. Drain_Control=Off 28 ==> Source_Flow=Medium Drain_Flow=Low 28   conf:(1)
70. Source_Flow=Medium Drain_Flow=Low 28 ==> Drain_Control=Off 28   conf:(1)
71. Source_Flow=Medium Drain_Control=Off 28 ==> Drain_Flow=Low 28   conf:(1)
72. Drain_Flow=Low Drain_Control=Off 28 ==> Source_Flow=Medium 28   conf:(1)
73. Drain_Control=Off 28 ==> Fault=None 28   conf:(1)
74. Drain_Flow=Low 28 ==> Fault=None 28   conf:(1)
75. Drain_Flow=Low 28 ==> Drain_Control=Off 28   conf:(1)
76. Drain_Control=Off 28 ==> Drain_Flow=Low 28   conf:(1)
77. Drain_Control=Off 28 ==> Source_Flow=Medium 28   conf:(1)
78. Drain_Flow=Low 28 ==> Source_Flow=Medium 28   conf:(1)
79. Sump_Level=Medium Drain_Flow=Low 27 ==> Source_Flow=Medium Drain_Control=Off Fault=None 27   conf:(1)
80. Sump_Level=Medium Drain_Control=Off 27 ==> Source_Flow=Medium Drain_Flow=Low Fault=None 27   conf:(1)
81. Source_Flow=Medium Sump_Level=Medium Drain_Flow=Low 27 ==> Drain_Control=Off Fault=None 27   conf:(1)
82. Source_Flow=Medium Sump_Level=Medium Drain_Control=Off 27 ==> Drain_Flow=Low Fault=None 27   conf:(1)

83. Sump_Level=Medium Drain_Flow=Low Drain_Control=Off 27 ==> Source_Flow=Medium Fault=None 27 conf:(1)
84. Sump_Level=Medium Drain_Flow=Low Fault=None 27 ==> Source_Flow=Medium Drain_Control=Off 27 conf:(1)
85. Sump_Level=Medium Drain_Control=Off Fault=None 27 ==> Source_Flow=Medium Drain_Flow=Low 27 conf:(1)
86. Source_Flow=Medium Sump_Level=Medium Drain_Flow=Low Drain_Control=Off 27 ==> Fault=None 27 conf:(1)
87. Source_Flow=Medium Sump_Level=Medium Drain_Flow=Low Fault=None 27 ==> Drain_Control=Off 27 conf:(1)
88. Source_Flow=Medium Sump_Level=Medium Drain_Control=Off Fault=None 27 ==> Drain_Flow=Low 27 conf:(1)
89. Sump_Level=Medium Drain_Flow=Low Drain_Control=Off Fault=None 27 ==> Source_Flow=Medium 27 conf:(1)
90. Sump_Level=Medium Drain_Flow=Low 27 ==> Drain_Control=Off Fault=None 27 conf:(1)
91. Sump_Level=Medium Drain_Control=Off 27 ==> Drain_Flow=Low Fault=None 27 conf:(1)
92. Sump_Level=Medium Drain_Flow=Low Drain_Control=Off 27 ==> Fault=None 27 conf:(1)
93. Sump_Level=Medium Drain_Flow=Low Fault=None 27 ==> Drain_Control=Off 27 conf:(1)
94. Sump_Level=Medium Drain_Control=Off Fault=None 27 ==> Drain_Flow=Low 27 conf:(1)
95. Heating_Control=Off 27 ==> Source_Flow=Medium Dispenser_Control=Off Fault=None 27 conf:(1)
96. Dispenser_Control=Off 27 ==> Source_Flow=Medium Heating_Control=Off Fault=None 27 conf:(1)
97. Source_Flow=Medium Heating_Control=Off 27 ==> Dispenser_Control=Off Fault=None 27 conf:(1)
98. Source_Flow=Medium Dispenser_Control=Off 27 ==> Heating_Control=Off Fault=None 27 conf:(1)
99. Heating_Control=Off Dispenser_Control=Off 27 ==> Source_Flow=Medium Fault=None 27 conf:(1)
100. Heating_Control=Off Fault=None 27 ==> Source_Flow=Medium Dispenser_Control=Off 27 conf:(1)

2] PREDICTIVEAPRIORI

=== Run information ===

Scheme: weka.associations.PredictiveApriori -N 100
Relation: dishwasher-weka.filters.unsupervised.attribute.Discretize-B40-M-1.0-R1
Instances: 40
Attributes: 17

Time
Source_Flow
WIV_Flow
WIV_Control
Sump_Level
Pump_Flow
Pump_Control
FWH_Flow
Heating_Control
NTC_Status
Flow_Switch_Status
Spray_Arm_Status
Dispenser_Control
Rinse_Aid_Control
Drain_Flow
Drain_Control
Fault

=== Associator model (full training set) ===

PredictiveApriori
===================

Best rules found:

1. Source_Flow=Medium 40 ==> Fault=None 40   acc:(0.98803)
2. Fault=None 40 ==> Source_Flow=Medium 40   acc:(0.98803)
3. Rinse_Aid_Control=Off 33 ==> Source_Flow=Medium Fault=None 33   acc:(0.98544)
4. WIV_Flow=Low 32 ==> Source_Flow=Medium WIV_Control=Off 32   acc:(0.98496)
5. WIV_Flow=Low 32 ==> Source_Flow=Medium Fault=None 32   acc:(0.98496)
6. WIV_Control=Off 32 ==> Source_Flow=Medium WIV_Flow=Low 32   acc:(0.98496)
7. WIV_Control=Off 32 ==> Source_Flow=Medium Fault=None 32   acc:(0.98496)
8. Sump_Level=Medium 32 ==> Source_Flow=Medium Fault=None 32   acc:(0.98496)
9. Source_Flow=Medium WIV_Flow=Low 32 ==> WIV_Control=Off Fault=None 32   acc:(0.98496)
10. Source_Flow=Medium WIV_Control=Off 32 ==> WIV_Flow=Low Fault=None 32   acc:(0.98496)
11. Drain_Flow=Low 28 ==> Source_Flow=Medium Drain_Control=Off 28   acc:(0.98261)
12. Drain_Flow=Low 28 ==> Source_Flow=Medium Fault=None 28   acc:(0.98261)
13. Drain_Control=Off 28 ==> Source_Flow=Medium Drain_Flow=Low 28   acc:(0.98261)
14. Drain_Control=Off 28 ==> Source_Flow=Medium Fault=None 28   acc:(0.98261)
15. Source_Flow=Medium Drain_Flow=Low 28 ==> Drain_Control=Off Fault=None 28   acc:(0.98261)
16. Source_Flow=Medium Drain_Control=Off 28 ==> Drain_Flow=Low Fault=None 28   acc:(0.98261)
17. Heating_Control=Off 27 ==> Source_Flow=Medium Dispenser_Control=Off 27   acc:(0.98189)
18. Heating_Control=Off 27 ==> Source_Flow=Medium Fault=None 27   acc:(0.98189)
19. Dispenser_Control=Off 27 ==> Source_Flow=Medium Heating_Control=Off 27   acc:(0.98189)
20. Dispenser_Control=Off 27 ==> Source_Flow=Medium Fault=None 27   acc:(0.98189)
21. Source_Flow=Medium Heating_Control=Off 27 ==> Dispenser_Control=Off Fault=None 27 acc:(0.98189)
22. Source_Flow=Medium Dispenser_Control=Off 27 ==> Heating_Control=Off Fault=None 27 acc:(0.98189)
23. Pump_Flow=Medium 25 ==> Source_Flow=Medium Sump_Level=Medium 25   acc:(0.98026)
24. Pump_Flow=Medium 25 ==> Source_Flow=Medium Pump_Control=On 25   acc:(0.98026)
25. Pump_Control=On 25 ==> Source_Flow=Medium Sump_Level=Medium 25   acc:(0.98026)
26. Pump_Control=On 25 ==> Source_Flow=Medium Pump_Flow=Medium 25   acc:(0.98026)
27. FWH_Flow=Medium 25 ==> Source_Flow=Medium Sump_Level=Medium 25   acc:(0.98026)
28. FWH_Flow=Medium 25 ==> Source_Flow=Medium Pump_Flow=Medium 25   acc:(0.98026)
29. Flow_Switch_Status=On 25 ==> Source_Flow=Medium WIV_Flow=Low 25   acc:(0.98026)
30. Flow_Switch_Status=On 25 ==> Source_Flow=Medium WIV_Control=Off 25   acc:(0.98026)
31. Spray_Arm_Status=Rotating 25 ==> Source_Flow=Medium Sump_Level=Medium 25   acc:(0.98026)
32. Spray_Arm_Status=Rotating 25 ==> Source_Flow=Medium Pump_Flow=Medium 25   acc:(0.98026)
33. Source_Flow=Medium Pump_Flow=Medium 25 ==> Sump_Level=Medium Spray_Arm_Status=Rotating 25   acc:(0.98026)
34. Source_Flow=Medium Pump_Flow=Medium 25 ==> Sump_Level=Medium Pump_Control=On 25 acc:(0.98026)
35. Source_Flow=Medium Pump_Control=On 25 ==> Sump_Level=Medium Spray_Arm_Status=Rotating 25 acc:(0.98026)
36. Source_Flow=Medium Pump_Control=On 25 ==> Sump_Level=Medium Pump_Flow=Medium 25 acc:(0.98026)
37. Source_Flow=Medium FWH_Flow=Medium 25 ==> Sump_Level=Medium Spray_Arm_Status=Rotating 25   acc:(0.98026)
38. Source_Flow=Medium FWH_Flow=Medium 25 ==> Sump_Level=Medium Pump_Flow=Medium 25 acc:(0.98026)
39. Source_Flow=Medium Flow_Switch_Status=On 25 ==> WIV_Flow=Low WIV_Control=Off 25 acc:(0.98026)
40. Source_Flow=Medium Flow_Switch_Status=On 25 ==> WIV_Flow=Low Sump_Level=Medium 25 acc:(0.98026)

41. Source_Flow=Medium Spray_Arm_Status=Rotating 25 ==> Sump_Level=Medium FWH_Flow=Medium 25 acc:(0.98026)
42. Source_Flow=Medium Spray_Arm_Status=Rotating 25 ==> Sump_Level=Medium Pump_Flow=Medium 25 acc:(0.98026)
43. WIV_Flow=Low Sump_Level=Medium 25 ==> Source_Flow=Medium Flow_Switch_Status=On 25 acc:(0.98026)
44. WIV_Flow=Low Sump_Level=Medium 25 ==> WIV_Control=Off Flow_Switch_Status=On 25 acc:(0.98026)
45. WIV_Flow=Low Flow_Switch_Status=On 25 ==> Source_Flow=Medium Sump_Level=Medium 25 acc:(0.98026)
46. WIV_Flow=Low Flow_Switch_Status=On 25 ==> WIV_Control=Off Sump_Level=Medium 25 acc:(0.98026)
47. WIV_Flow=Low Rinse_Aid_Control=Off 25 ==> WIV_Control=Off Fault=None 25 acc:(0.98026)
48. WIV_Control=Off Sump_Level=Medium 25 ==> Source_Flow=Medium Flow_Switch_Status=On 25 acc:(0.98026)
49. WIV_Control=Off Sump_Level=Medium 25 ==> WIV_Flow=Low Flow_Switch_Status=On 25 acc:(0.98026)
50. WIV_Control=Off Flow_Switch_Status=On 25 ==> Source_Flow=Medium Sump_Level=Medium 25 acc:(0.98026)
51. WIV_Control=Off Flow_Switch_Status=On 25 ==> WIV_Flow=Low Sump_Level=Medium 25 acc:(0.98026)
52. WIV_Control=Off Rinse_Aid_Control=Off 25 ==> WIV_Flow=Low Fault=None 25 acc:(0.98026)
53. Sump_Level=Medium Pump_Flow=Medium 25 ==> Source_Flow=Medium Spray_Arm_Status=Rotating 25 acc:(0.98026)
54. Sump_Level=Medium Pump_Flow=Medium 25 ==> Source_Flow=Medium FWH_Flow=Medium 25 acc:(0.98026)
55. Sump_Level=Medium Pump_Control=On 25 ==> Source_Flow=Medium Spray_Arm_Status=Rotating 25 acc:(0.98026)
56. Sump_Level=Medium Pump_Control=On 25 ==> Source_Flow=Medium FWH_Flow=Medium 25 acc:(0.98026)
57. Sump_Level=Medium FWH_Flow=Medium 25 ==> Source_Flow=Medium Spray_Arm_Status=Rotating 25 acc:(0.98026)
58. NTC_Status=In_Range 23 ==> Source_Flow=Medium Rinse_Aid_Control=Off 23 acc:(0.97829)
59. NTC_Status=In_Range 23 ==> Source_Flow=Medium Fault=None 23 acc:(0.97829)
60. Source_Flow=Medium NTC_Status=In_Range 23 ==> Rinse_Aid_Control=Off Fault=None 23 acc:(0.97829)
61. WIV_Flow=Low Pump_Flow=Medium 20 ==> FWH_Flow=Medium 20 acc:(0.9745)
62. NTC_Status=Cut_Off 17 ==> Source_Flow=Medium WIV_Flow=Low 17 acc:(0.96922)
63. NTC_Status=Cut_Off 17 ==> Source_Flow=Medium WIV_Control=Off 17 acc:(0.96922)
64. Source_Flow=Medium NTC_Status=Cut_Off 17 ==> WIV_Flow=Low Dispenser_Control=Off 17 acc:(0.96922)
65. Source_Flow=Medium NTC_Status=Cut_Off 17 ==> WIV_Flow=Low WIV_Control=Off 17 acc:(0.96922)
66. Pump_Flow=Low 15 ==> Source_Flow=Medium Pump_Control=Off 15 acc:(0.96444)
67. Pump_Flow=Low 15 ==> Source_Flow=Medium FWH_Flow=Low 15 acc:(0.96444)
68. Pump_Control=Off 15 ==> Source_Flow=Medium Pump_Flow=Low 15 acc:(0.96444)
69. Pump_Control=Off 15 ==> Source_Flow=Medium FWH_Flow=Low 15 acc:(0.96444)
70. FWH_Flow=Low 15 ==> Source_Flow=Medium Pump_Flow=Low 15 acc:(0.96444)
71. FWH_Flow=Low 15 ==> Source_Flow=Medium Pump_Control=Off 15 acc:(0.96444)
72. Flow_Switch_Status=Off 15 ==> Source_Flow=Medium Heating_Control=Off 15 acc:(0.96444)
73. Flow_Switch_Status=Off 15 ==> Source_Flow=Medium Dispenser_Control=Off 15 acc:(0.96444)
74. Spray_Arm_Status=Not_Rotating 15 ==> Source_Flow=Medium Pump_Flow=Low 15 acc:(0.96444)
75. Spray_Arm_Status=Not_Rotating 15 ==> Source_Flow=Medium Pump_Control=Off 15 acc:(0.96444)
76. Source_Flow=Medium Pump_Flow=Low 15 ==> Pump_Control=Off Spray_Arm_Status=Not_Rotating 15 acc:(0.96444)

77. Source_Flow=Medium Pump_Flow=Low 15 ==> Pump_Control=Off FWH_Flow=Low 15 acc:(0.96444)
78. Source_Flow=Medium Pump_Control=Off 15 ==> Pump_Flow=Low Spray_Arm_Status=Not_Rotating 15 acc:(0.96444)
79. Source_Flow=Medium Pump_Control=Off 15 ==> Pump_Flow=Low FWH_Flow=Low 15 acc:(0.96444)
80. Source_Flow=Medium FWH_Flow=Low 15 ==> Pump_Flow=Low Spray_Arm_Status=Not_Rotating 15 acc:(0.96444)
81. Source_Flow=Medium FWH_Flow=Low 15 ==> Pump_Flow=Low Pump_Control=Off 15 acc:(0.96444)
82. Source_Flow=Medium Flow_Switch_Status=Off 15 ==> Heating_Control=Off Dispenser_Control=Off 15 acc:(0.96444)
83. Source_Flow=Medium Flow_Switch_Status=Off 15 ==> Heating_Control=Off Rinse_Aid_Control=Off 15 acc:(0.96444)
84. Source_Flow=Medium Spray_Arm_Status=Not_Rotating 15 ==> Pump_Flow=Low Heating_Control=Off 15 acc:(0.96444)
85. Source_Flow=Medium Spray_Arm_Status=Not_Rotating 15 ==> Pump_Flow=Low Pump_Control=Off 15 acc:(0.96444)
86. Heating_Control=On 13 ==> Source_Flow=Medium WIV_Flow=Low 13 acc:(0.95814)
87. Heating_Control=On 13 ==> Source_Flow=Medium WIV_Control=Off 13 acc:(0.95814)
88. Dispenser_Control=On 13 ==> Source_Flow=Medium WIV_Flow=Low 13 acc:(0.95814)
89. Dispenser_Control=On 13 ==> Source_Flow=Medium WIV_Control=Off 13 acc:(0.95814)
90. Source_Flow=Medium Heating_Control=On 13 ==> WIV_Flow=Low Pump_Flow=Medium 13 acc:(0.95814)
91. Source_Flow=Medium Heating_Control=On 13 ==> WIV_Flow=Low WIV_Control=Off 13 acc:(0.95814)
92. Source_Flow=Medium Dispenser_Control=On 13 ==> WIV_Flow=Low Pump_Flow=Medium 13 acc:(0.95814)
93. Source_Flow=Medium Dispenser_Control=On 13 ==> WIV_Flow=Low WIV_Control=Off 13 acc:(0.95814)
94. Drain_Flow=Medium 12 ==> Source_Flow=Medium WIV_Flow=Low 12 acc:(0.9542)
95. Drain_Flow=Medium 12 ==> Source_Flow=Medium WIV_Control=Off 12 acc:(0.9542)
96. Drain_Control=On 12 ==> Source_Flow=Medium WIV_Flow=Low 12 acc:(0.9542)
97. Drain_Control=On 12 ==> Source_Flow=Medium WIV_Control=Off 12 acc:(0.9542)
98. Drain_Flow=Low 28 ==> Source_Flow=Medium Sump_Level=Medium 27 acc:(0.94299)
99. Drain_Flow=Low 28 ==> Sump_Level=Medium Drain_Control=Off 27 acc:(0.94299)
100. Drain_Control=Off 28 ==> Source_Flow=Medium Sump_Level=Medium 27 acc:(0.94299)

3] Tertius

=== Run information ===

Scheme: weka.associations.Tertius -K 10 -F 0.0 -C 0.0 -N 1.0 -L 4 -G 0 -c 0 -I 0 -O -p -P 0
Relation: dishwasher-weka.filters.unsupervised.attribute.Discretize-B40-M-1.0-R1
Instances: 40
Attributes: 17
Time
Source_Flow
WIV_Flow
WIV_Control
Sump_Level
Pump_Flow
Pump_Control
FWH_Flow

Heating_Control
NTC_Status
Flow_Switch_Status
Spray_Arm_Status
Dispenser_Control
Rinse_Aid_Control
Drain_Flow
Drain_Control
Fault
=== Associator model (full training set) ===

Tertius
=======

1. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> Pump_Flow = Low or Flow_Switch_Status = Off
2. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> Pump_Flow = Low or Flow_Switch_Status = Off
3. /* 1.000000 1.000000 0.000000 */ Pump_Flow = Medium and Flow_Switch_Status = On ==> Heating_Control = On or Rinse_Aid_Control = On
4. /* 1.000000 1.000000 0.000000 */ Pump_Flow = Medium and Flow_Switch_Status = On ==> Dispenser_Control = On or Rinse_Aid_Control = On
5. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> Pump_Control = Off or Flow_Switch_Status = Off
6. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> Pump_Control = Off or Flow_Switch_Status = Off
7. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> FWH_Flow = Low or Flow_Switch_Status = Off
8. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> FWH_Flow = Low or Flow_Switch_Status = Off
9. /* 1.000000 1.000000 0.000000 */ FWH_Flow = Medium and Flow_Switch_Status = On ==> Heating_Control = On or Rinse_Aid_Control = On
10. /* 1.000000 1.000000 0.000000 */ FWH_Flow = Medium and Flow_Switch_Status = On ==> Dispenser_Control = On or Rinse_Aid_Control = On
11. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> Flow_Switch_Status = Off or Drain_Flow = Medium
12. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> Flow_Switch_Status = Off or Drain_Control = On
13. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> Flow_Switch_Status = Off or Spray_Arm_Status = Not_Rotating
14. /* 1.000000 1.000000 0.000000 */ Flow_Switch_Status = On and Drain_Flow = Low ==> Dispenser_Control = On or Rinse_Aid_Control = On
15. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> Flow_Switch_Status = Off or Drain_Flow = Medium
16. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> Flow_Switch_Status = Off or Drain_Control = On
17. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> Flow_Switch_Status = Off or Spray_Arm_Status = Not_Rotating
18. /* 1.000000 1.000000 0.000000 */ Flow_Switch_Status = On and Drain_Flow = Low ==> Heating_Control = On or Rinse_Aid_Control = On
19. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and Pump_Flow = Medium ==> Heating_Control = On or Rinse_Aid_Control = On
20. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and Pump_Control = On ==> Heating_Control = On or Rinse_Aid_Control = On

21. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and FWH_Flow = Medium ==> Heating_Control = On or Rinse_Aid_Control = On
22. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and Spray_Arm_Status = Rotating ==> Heating_Control = On or Rinse_Aid_Control = On
23. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and Drain_Flow = Low ==> Heating_Control = On or Rinse_Aid_Control = On
24. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and Drain_Control = Off ==> Heating_Control = On or Rinse_Aid_Control = On
25. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and Pump_Flow = Medium ==> Dispenser_Control = On or Rinse_Aid_Control = On
26. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and Pump_Control = On ==> Dispenser_Control = On or Rinse_Aid_Control = On
27. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and FWH_Flow = Medium ==> Dispenser_Control = On or Rinse_Aid_Control = On
28. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and Spray_Arm_Status = Rotating ==> Dispenser_Control = On or Rinse_Aid_Control = On
29. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and Drain_Flow = Low ==> Dispenser_Control = On or Rinse_Aid_Control = On
30. /* 1.000000 1.000000 0.000000 */ WIV_Flow = Low and Drain_Control = Off ==> Dispenser_Control = On or Rinse_Aid_Control = On
31. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or Pump_Flow = Low
32. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or Pump_Control = Off
33. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or FWH_Flow = Low
34. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or Spray_Arm_Status = Not_Rotating
35. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or Drain_Flow = Medium
36. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or Drain_Control = On
37. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or Pump_Flow = Low
38. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or Pump_Control = Off
39. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or FWH_Flow = Low
40. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or Spray_Arm_Status = Not_Rotating
41. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or Drain_Flow = Medium
42. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> WIV_Flow = Medium or Drain_Control = On
43. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and Pump_Flow = Medium ==> Heating_Control = On or Rinse_Aid_Control = On
44. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and Pump_Control = On ==> Heating_Control = On or Rinse_Aid_Control = On
45. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and FWH_Flow = Medium ==> Heating_Control = On or Rinse_Aid_Control = On
46. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and Spray_Arm_Status = Rotating ==> Heating_Control = On or Rinse_Aid_Control = On
47. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and Drain_Flow = Low ==> Heating_Control = On or Rinse_Aid_Control = On
48. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and Drain_Control = Off ==> Heating_Control = On or Rinse_Aid_Control = On

49. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and Pump_Flow = Medium ==> Dispenser_Control = On or Rinse_Aid_Control = On
50. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and Pump_Control = On ==> Dispenser_Control = On or Rinse_Aid_Control = On
51. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and FWH_Flow = Medium ==> Dispenser_Control = On or Rinse_Aid_Control = On
52. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and Spray_Arm_Status = Rotating ==> Dispenser_Control = On or Rinse_Aid_Control = On
53. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and Drain_Flow = Low ==> Dispenser_Control = On or Rinse_Aid_Control = On
54. /* 1.000000 1.000000 0.000000 */ WIV_Control = Off and Drain_Control = Off ==> Dispenser_Control = On or Rinse_Aid_Control = On
55. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> WIV_Control = On or Pump_Flow = Low
56. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> WIV_Control = On or FWH_Flow = Low
57. /* 1.000000 1.000000 0.000000 */ Heating_Control = Off and Rinse_Aid_Control = Off ==> WIV_Control = On or Drain_Flow = Medium
58. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> WIV_Control = On or Pump_Flow = Low
59. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> WIV_Control = On or FWH_Flow = Low
60. /* 1.000000 1.000000 0.000000 */ Dispenser_Control = Off and Rinse_Aid_Control = Off ==> WIV_Control = On or Drain_Flow = Medium
61. /* 0.997502 1.000000 0.000000 */ WIV_Flow = Low and Heating_Control = Off ==> NTC_Status = Cut_Off or Drain_Flow = Medium
62. /* 0.997502 1.000000 0.000000 */ WIV_Flow = Low and Heating_Control = Off ==> NTC_Status = Cut_Off or Drain_Control = On
63. /* 0.997502 1.000000 0.000000 */ WIV_Flow = Low and Dispenser_Control = Off ==> NTC_Status = Cut_Off or Drain_Flow = Medium
64. /* 0.997502 1.000000 0.000000 */ WIV_Flow = Low and Dispenser_Control = Off ==> NTC_Status = Cut_Off or Drain_Control = On
65. /* 0.997502 1.000000 0.000000 */ NTC_Status = In_Range and Drain_Flow = Low ==> WIV_Flow = Medium or Heating_Control = On
66. /* 0.997502 1.000000 0.000000 */ NTC_Status = In_Range and Drain_Control = Off ==> WIV_Flow = Medium or Heating_Control = On
67. /* 0.997502 1.000000 0.000000 */ NTC_Status = In_Range and Drain_Flow = Low ==> WIV_Flow = Medium or Dispenser_Control = On
68. /* 0.997502 1.000000 0.000000 */ NTC_Status = In_Range and Drain_Control = Off ==> WIV_Flow = Medium or Dispenser_Control = On
69. /* 0.997502 1.000000 0.000000 */ WIV_Control = Off and Heating_Control = Off ==> NTC_Status = Cut_Off or Drain_Flow = Medium
70. /* 0.997502 1.000000 0.000000 */ WIV_Control = Off and Heating_Control = Off ==> NTC_Status = Cut_Off or Drain_Control = On
71. /* 0.997502 1.000000 0.000000 */ WIV_Control = Off and Dispenser_Control = Off ==> NTC_Status = Cut_Off or Drain_Flow = Medium
72. /* 0.997502 1.000000 0.000000 */ WIV_Control = Off and Dispenser_Control = Off ==> NTC_Status = Cut_Off or Drain_Control = On
73. /* 0.997502 1.000000 0.000000 */ NTC_Status = In_Range and Drain_Flow = Low ==> WIV_Control = On or Heating_Control = On
74. /* 0.997502 1.000000 0.000000 */ NTC_Status = In_Range and Drain_Flow = Low ==> WIV_Control = On or Dispenser_Control = On
75. /* 0.997502 1.000000 0.000000 */ WIV_Flow = Low and Heating_Control = Off ==> Rinse_Aid_Control = On or Drain_Flow = Medium
76. /* 0.997502 1.000000 0.000000 */ WIV_Flow = Low and Dispenser_Control = Off ==> Rinse_Aid_Control = On or Drain_Flow = Medium

77. /* 0.997502 1.000000 0.000000 */ Rinse_Aid_Control = Off and Drain_Flow = Low ==> WIV_Flow = Medium or Heating_Control = On
78. /* 0.997502 1.000000 0.000000 */ Rinse_Aid_Control = Off and Drain_Flow = Low ==> WIV_Flow = Medium or Dispenser_Control = On
79. /* 0.950609 0.950000 0.000000 */ WIV_Flow = Low and Dispenser_Control = Off ==> Sump_Level = Low or NTC_Status = Cut_Off
80. /* 0.950609 0.952381 0.000000 */ Sump_Level = Medium and NTC_Status = In_Range ==> WIV_Flow = Medium or Dispenser_Control = On
81. /* 0.950609 0.950000 0.000000 */ WIV_Control = Off and Dispenser_Control = Off ==> Sump_Level = Low or NTC_Status = Cut_Off
82. /* 0.950609 0.952381 0.000000 */ Sump_Level = Medium and NTC_Status = In_Range ==> WIV_Control = On or Dispenser_Control = On
83. /* 0.950609 0.950000 0.000000 */ WIV_Flow = Low and Heating_Control = Off ==> Sump_Level = Low or NTC_Status = Cut_Off
84. /* 0.950609 0.952381 0.000000 */ Sump_Level = Medium and NTC_Status = In_Range ==> WIV_Flow = Medium or Heating_Control = On
85. /* 0.950609 0.950000 0.000000 */ WIV_Control = Off and Heating_Control = Off ==> Sump_Level = Low or NTC_Status = Cut_Off
86. /* 0.950609 0.952381 0.000000 */ Sump_Level = Medium and NTC_Status = In_Range ==> WIV_Control = On or Heating_Control = On
87. /* 0.945468 1.000000 0.045455 */ WIV_Flow = Low and Dispenser_Control = Off ==> Time = '(-inf-1.975]' or NTC_Status = Cut_Off
88. /* 0.945468 1.000000 0.045455 */ WIV_Control = Off and Dispenser_Control = Off ==> Time = '(-inf-1.975]' or NTC_Status = Cut_Off
89. /* 0.945468 1.000000 0.045455 */ WIV_Flow = Low and Dispenser_Control = Off ==> Time = '(1.975-2.95]' or NTC_Status = Cut_Off
90. /* 0.945468 1.000000 0.045455 */ WIV_Control = Off and Dispenser_Control = Off ==> Time = '(1.975-2.95]' or NTC_Status = Cut_Off
91. /* 0.945468 1.000000 0.045455 */ WIV_Flow = Low and Heating_Control = Off ==> Time = '(-inf-1.975]' or NTC_Status = Cut_Off
92. /* 0.945468 1.000000 0.045455 */ WIV_Control = Off and Heating_Control = Off ==> Time = '(-inf-1.975]' or NTC_Status = Cut_Off
93. /* 0.945468 1.000000 0.045455 */ WIV_Flow = Low and Heating_Control = Off ==> Time = '(1.975-2.95]' or NTC_Status = Cut_Off
94. /* 0.945468 1.000000 0.045455 */ WIV_Control = Off and Heating_Control = Off ==> Time = '(1.975-2.95]' or NTC_Status = Cut_Off
95. /* 0.938446 1.000000 0.000000 */ Pump_Control = Off ==> Pump_Flow = Low
96. /* 0.938446 1.000000 0.000000 */ FWH_Flow = Low ==> Pump_Flow = Low
97. /* 0.938446 1.000000 0.000000 */ Spray_Arm_Status = Not_Rotating ==> Pump_Flow = Low
98. /* 0.938446 1.000000 0.000000 */ Pump_Flow = Medium ==> Pump_Control = On
99. /* 0.938446 1.000000 0.000000 */ Pump_Flow = Medium ==> FWH_Flow = Medium
100. /* 0.938446 1.000000 0.000000 */ Pump_Flow = Medium ==> Spray_Arm_Status = Rotating
101. /* 0.938446 1.000000 0.000000 */ FWH_Flow = Low ==> Pump_Control = Off
102. /* 0.938446 1.000000 0.000000 */ Spray_Arm_Status = Not_Rotating ==> Pump_Control = Off
103. /* 0.938446 1.000000 0.000000 */ Pump_Control = On ==> FWH_Flow = Medium
104. /* 0.938446 1.000000 0.000000 */ Spray_Arm_Status = Not_Rotating ==> FWH_Flow = Low
105. /* 0.938446 1.000000 0.000000 */ FWH_Flow = Medium ==> Spray_Arm_Status = Rotating
106. /* 0.938446 1.000000 0.000000 */ WIV_Flow = Low and Sump_Level = Medium ==> Flow_Switch_Status = On
107. /* 0.938446 1.000000 0.000000 */ Flow_Switch_Status = Off ==> WIV_Flow = Medium or Sump_Level = Low
108. /* 0.938446 1.000000 0.000000 */ WIV_Control = Off and Sump_Level = Medium ==> Flow_Switch_Status = On
109. /* 0.938446 1.000000 0.000000 */ Flow_Switch_Status = Off ==> WIV_Control = On or Sump_Level = Low
110. /* 0.938446 1.000000 0.000000 */ Pump_Flow = Low ==> Pump_Control = Off

111. /* 0.938446 1.000000 0.000000 */ Pump_Flow = Low ==> FWH_Flow = Low
112. /* 0.938446 1.000000 0.000000 */ Pump_Flow = Low ==> Spray_Arm_Status = Not_Rotating
113. /* 0.938446 1.000000 0.000000 */ Pump_Control = On ==> Pump_Flow = Medium
114. /* 0.938446 1.000000 0.000000 */ FWH_Flow = Medium ==> Pump_Flow = Medium
115. /* 0.938446 1.000000 0.000000 */ Spray_Arm_Status = Rotating ==> Pump_Flow = Medium
116. /* 0.938446 1.000000 0.000000 */ Pump_Control = Off ==> FWH_Flow = Low
117. /* 0.938446 1.000000 0.000000 */ Pump_Control = Off ==> Spray_Arm_Status = Not_Rotating
118. /* 0.938446 1.000000 0.000000 */ FWH_Flow = Medium ==> Pump_Control = On
119. /* 0.938446 1.000000 0.000000 */ FWH_Flow = Low ==> Spray_Arm_Status = Not_Rotating
120. /* 0.938446 1.000000 0.000000 */ Spray_Arm_Status = Rotating ==> FWH_Flow = Medium
121. /* 0.935282 1.000000 0.055556 */ NTC_Status = In_Range ==> Time = '(-inf-1.975]' or WIV_Flow = Medium or Heating_Control = On
122. /* 0.935282 1.000000 0.055556 */ NTC_Status = In_Range ==> Time = '(-inf-1.975]' or WIV_Flow = Medium or Dispenser_Control = On
123. /* 0.935282 1.000000 0.055556 */ NTC_Status = In_Range ==> Time = '(1.975-2.95]' or WIV_Flow = Medium or Heating_Control = On
124. /* 0.935282 1.000000 0.055556 */ NTC_Status = In_Range ==> Time = '(1.975-2.95]' or WIV_Flow = Medium or Dispenser_Control = On
125. /* 0.895236 0.894737 0.000000 */ NTC_Status = Cut_Off ==> Rinse_Aid_Control = On or Drain_Flow = Medium
126. /* 0.895236 0.894737 0.000000 */ NTC_Status = Cut_Off ==> Rinse_Aid_Control = On or Drain_Control = On
127. /* 0.895236 0.913043 0.000000 */ Rinse_Aid_Control = Off and Drain_Flow = Low ==> NTC_Status = In_Range
128. /* 0.894329 1.000000 0.086957 */ WIV_Flow = Low and Heating_Control = Off ==> NTC_Status = Cut_Off
129. /* 0.894329 1.000000 0.086957 */ WIV_Flow = Low and Dispenser_Control = Off ==> NTC_Status = Cut_Off
130. /* 0.894329 1.000000 0.105263 */ NTC_Status = In_Range ==> WIV_Flow = Medium or Heating_Control = On
131. /* 0.894329 1.000000 0.105263 */ NTC_Status = In_Range ==> WIV_Flow = Medium or Dispenser_Control = On
132. /* 0.894329 1.000000 0.086957 */ WIV_Control = Off and Heating_Control = Off ==> NTC_Status = Cut_Off
133. /* 0.894329 1.000000 0.086957 */ WIV_Control = Off and Dispenser_Control = Off ==> NTC_Status = Cut_Off
134. /* 0.881025 1.000000 0.000000 */ Dispenser_Control = Off ==> NTC_Status = Cut_Off or Flow_Switch_Status = Off
135. /* 0.881025 1.000000 0.000000 */ Heating_Control = Off ==> Dispenser_Control = Off
136. /* 0.881025 1.000000 0.000000 */ Heating_Control = Off ==> NTC_Status = Cut_Off or Flow_Switch_Status = Off
137. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and Pump_Flow = Medium and NTC_Status = In_Range ==> Heating_Control = On
138. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and Pump_Control = On and NTC_Status = In_Range ==> Heating_Control = On
139. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and FWH_Flow = Medium and NTC_Status = In_Range ==> Heating_Control = On
140. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and Sump_Level = Medium and NTC_Status = In_Range ==> Heating_Control = On
141. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and Pump_Flow = Medium and NTC_Status = In_Range ==> Dispenser_Control = On
142. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and Pump_Control = On and NTC_Status = In_Range ==> Dispenser_Control = On
143. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and FWH_Flow = Medium and NTC_Status = In_Range ==> Dispenser_Control = On

144. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and Sump_Level = Medium and NTC_Status = In_Range ==> Dispenser_Control = On
145. /* 0.881025 1.000000 0.000000 */ Heating_Control = Off ==> WIV_Flow = Medium or Pump_Flow = Low or NTC_Status = Cut_Off
146. /* 0.881025 1.000000 0.000000 */ Heating_Control = Off ==> WIV_Flow = Medium or Pump_Control = Off or NTC_Status = Cut_Off
147. /* 0.881025 1.000000 0.000000 */ Heating_Control = Off ==> WIV_Flow = Medium or FWH_Flow = Low or NTC_Status = Cut_Off
148. /* 0.881025 1.000000 0.000000 */ Heating_Control = Off ==> WIV_Flow = Medium or Sump_Level = Low or NTC_Status = Cut_Off
149. /* 0.881025 1.000000 0.000000 */ Dispenser_Control = Off ==> WIV_Flow = Medium or Pump_Flow = Low or NTC_Status = Cut_Off
150. /* 0.881025 1.000000 0.000000 */ Dispenser_Control = Off ==> WIV_Flow = Medium or Pump_Control = Off or NTC_Status = Cut_Off
151. /* 0.881025 1.000000 0.000000 */ Dispenser_Control = Off ==> WIV_Flow = Medium or FWH_Flow = Low or NTC_Status = Cut_Off
152. /* 0.881025 1.000000 0.000000 */ Dispenser_Control = Off ==> WIV_Flow = Medium or Sump_Level = Low or NTC_Status = Cut_Off
153. /* 0.881025 1.000000 0.000000 */ WIV_Control = Off and Pump_Flow = Medium and NTC_Status = In_Range ==> Heating_Control = On
154. /* 0.881025 1.000000 0.000000 */ WIV_Control = Off and Pump_Control = On and NTC_Status = In_Range ==> Heating_Control = On
155. /* 0.881025 1.000000 0.000000 */ WIV_Control = Off and FWH_Flow = Medium and NTC_Status = In_Range ==> Heating_Control = On
156. /* 0.881025 1.000000 0.000000 */ WIV_Control = Off and Pump_Flow = Medium and NTC_Status = In_Range ==> Dispenser_Control = On
157. /* 0.881025 1.000000 0.000000 */ WIV_Control = Off and Pump_Control = On and NTC_Status = In_Range ==> Dispenser_Control = On
158. /* 0.881025 1.000000 0.000000 */ WIV_Control = Off and FWH_Flow = Medium and NTC_Status = In_Range ==> Dispenser_Control = On
159. /* 0.881025 1.000000 0.000000 */ Heating_Control = Off ==> WIV_Control = On or Pump_Flow = Low or NTC_Status = Cut_Off
160. /* 0.881025 1.000000 0.000000 */ Heating_Control = Off ==> WIV_Control = On or FWH_Flow = Low or NTC_Status = Cut_Off
161. /* 0.881025 1.000000 0.000000 */ Dispenser_Control = Off ==> WIV_Control = On or Pump_Flow = Low or NTC_Status = Cut_Off
162. /* 0.881025 1.000000 0.000000 */ Dispenser_Control = Off ==> WIV_Control = On or FWH_Flow = Low or NTC_Status = Cut_Off
163. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and NTC_Status = In_Range and Spray_Arm_Status = Rotating ==> Heating_Control = On
164. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and NTC_Status = In_Range and Drain_Flow = Low ==> Heating_Control = On
165. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and NTC_Status = In_Range and Spray_Arm_Status = Rotating ==> Dispenser_Control = On
166. /* 0.881025 1.000000 0.000000 */ WIV_Flow = Low and NTC_Status = In_Range and Drain_Flow = Low ==> Dispenser_Control = On
167. /* 0.881025 1.000000 0.000000 */ Heating_Control = Off ==> WIV_Flow = Medium or NTC_Status = Cut_Off or Spray_Arm_Status = Not_Rotating
168. /* 0.881025 1.000000 0.000000 */ Heating_Control = Off ==> WIV_Flow = Medium or NTC_Status = Cut_Off or Drain_Flow = Medium
169. /* 0.881025 1.000000 0.000000 */ Dispenser_Control = Off ==> WIV_Flow = Medium or NTC_Status = Cut_Off or Spray_Arm_Status = Not_Rotating
170. /* 0.881025 1.000000 0.000000 */ Dispenser_Control = Off ==> WIV_Flow = Medium or NTC_Status = Cut_Off or Drain_Flow = Medium
171. /* 0.881025 1.000000 0.000000 */ WIV_Control = Off and NTC_Status = In_Range and Spray_Arm_Status = Rotating ==> Heating_Control = On

172. /* 0.881025 1.000000 0.000000 */ WIV_Control = Off and NTC_Status = In_Range and Spray_Arm_Status = Rotating ==> Dispenser_Control = On
173. /* 0.881025 1.000000 0.000000 */ Dispenser_Control = Off ==> Heating_Control = Off
174. /* 0.859881 0.863636 0.000000 */ WIV_Flow = Low and Heating_Control = Off ==> Pump_Flow = Low or NTC_Status = Cut_Off

Number of hypotheses considered: 741472
Number of hypotheses explored: 315482
Time: 01 min 50 s 739 ms

V Bosch Dishwasher Model No. SHX56C02 Documents

Blocks of Dishwasher

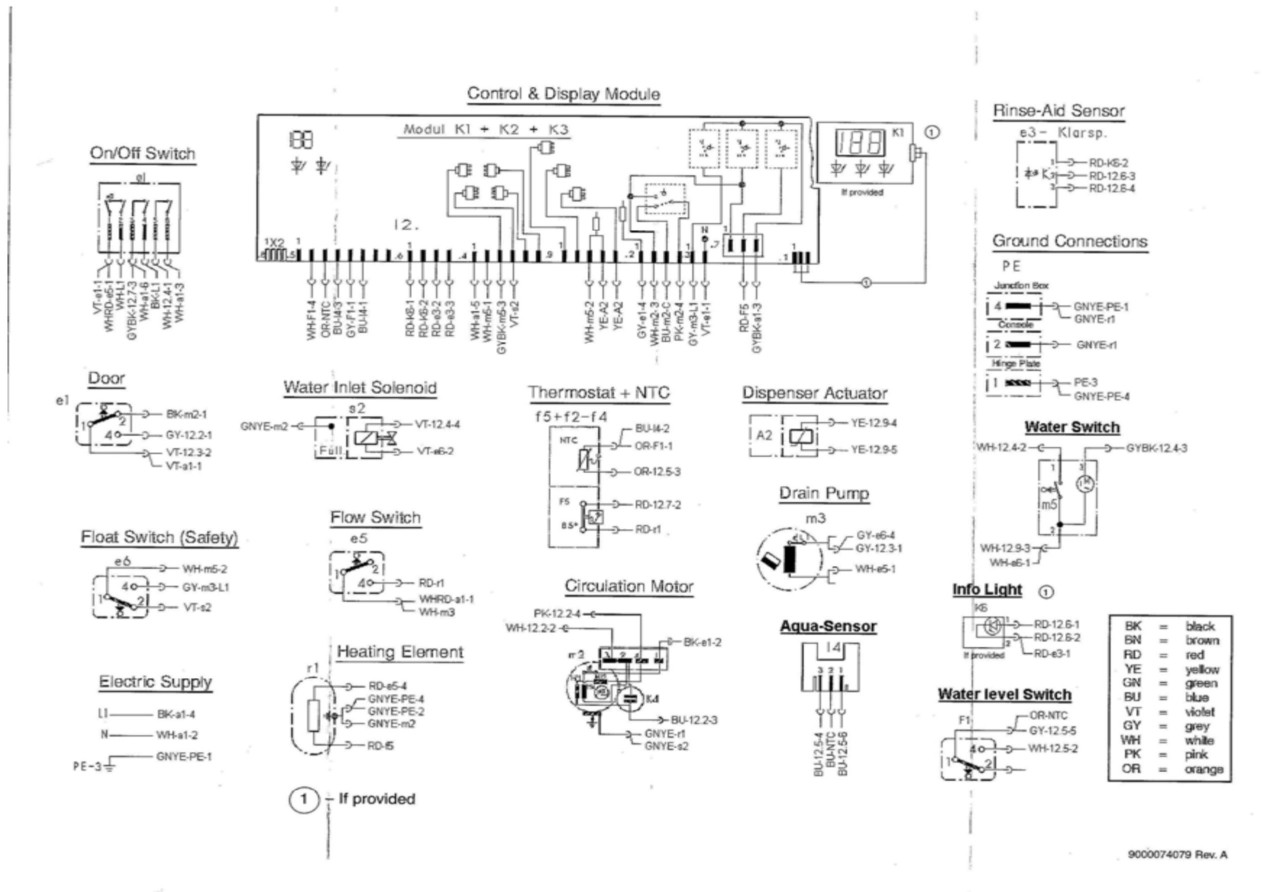


Figure 34 Bosch dishwasher blocks

Connection of Dishwasher

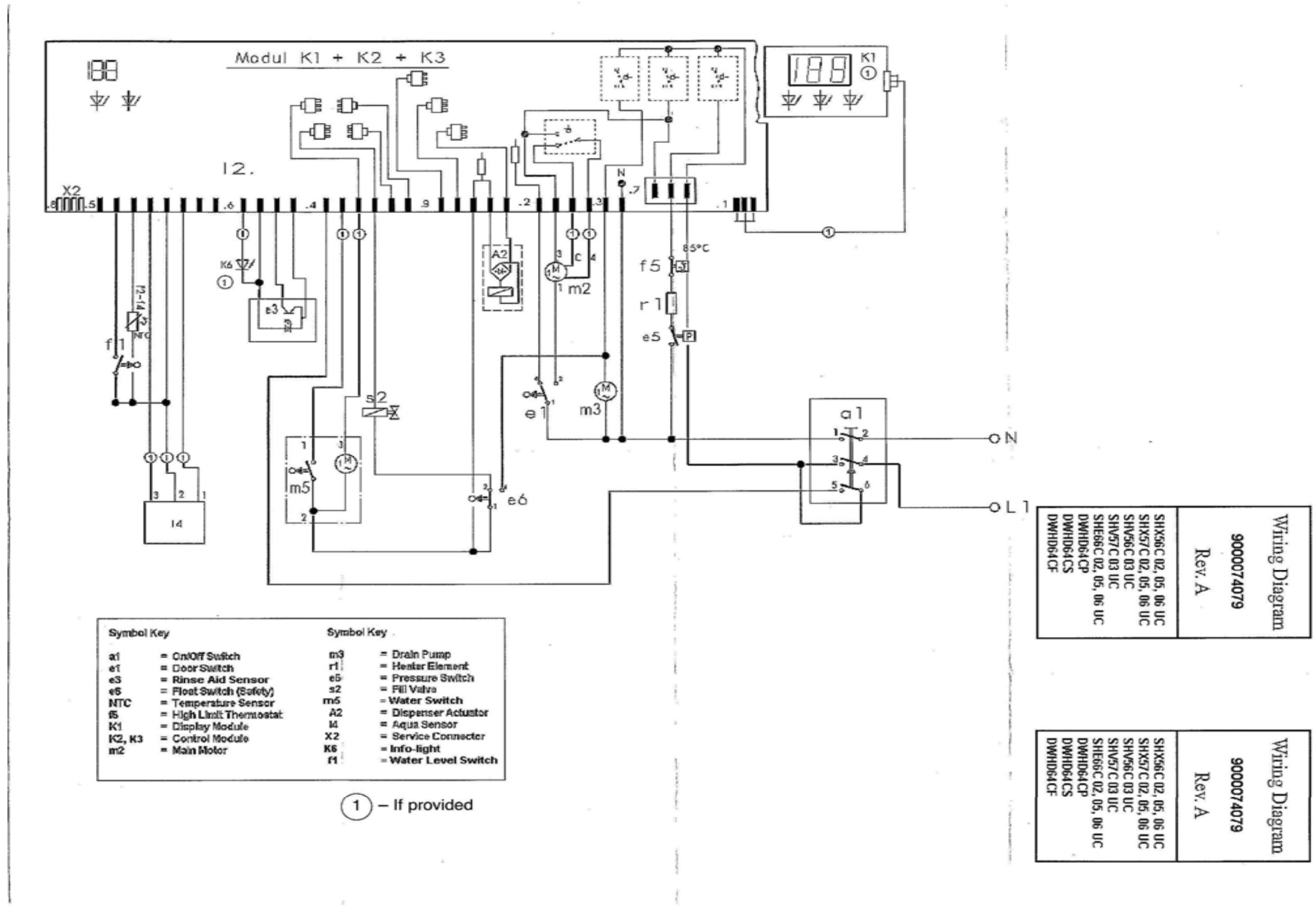


Figure 35 Bosch Dishwasher connection diagram

Customer Service Program Algorithm

KD Prüfprogramm Dokumentation: 56000000120478 / Rev, Seq: A, 1 / Date: 17.05.2005 PG / ESS Knaus

Sonderprogramm Kundendienst TF

Mat.Nr. 9000.058.297

INDEX	Funktion	Temperatur	Zeit [s]	Sensor	Füllmenge	WS gedr.	OK	UK	Bemerkung
						immer ein	immer ein	immer ein	
	P_z		15						
	FW_m				1,5L				
	PA_z		5						
	U_FW_m				3,0L/2,8L				
	U_H_ZR_z		120						
	U_H_t	60							
	U_ZK_z		120						
	U_H_ASK_0								
	SP_z		30						
	P_z		45						
	U_P_z		15						

Switch into the Customer- Service- Programme

While pressing two buttons turn the machine on with the main switch. The coding of the control unit will be shown in the display or with the LED's (bit- coded) until you release the buttons.

After releasing the buttons you're at the Special- Function- Menu
Display on the facia panel:

- LED B glow, LED C blinks
- If LED C isn't glowing or the Display doesn't show P1 press Button B many times
- pressing the Button C will start the Customer- Service- Programme
- pressing the Button B will jump to the next programme step
- blinking LED's (2Hz) display a detected error (wtih 7- segment e.g.: E2)

Programme interuption:

The programme can only be interrupted by pressing the main switch (overtravel contact detection)

Errorcodes (showed with LED's and/or Display in the test programme)

	LED's A	LED's B	LED's C	7- Segment	Error Type	Priority
	○	○	●	E1	Heating Error	high
	○	●	○	E2	NTC- Error	
	○	●	●	E3	Filling Error	
	●	○	○	E4	Waterswitch	
	●	○	●	E5	Tacho / Motor Error	
	●	●	○	E6	Aquasensor Error	
	●	●	●	E7	Water detected base sump	
●	○	○	●	E8	Heating Protection	↓
●	●	●	●	EF	Water detected base	low

○ LED dark
● LED flash

Kundendienstpruefprogramm ~~1KD630UC_G__FTE_01~~

Figure 36 Customer Service Program Algorithm

Program Notations

Actions

U_z	1
P_z	2
PA_z	3
AWT_z	4
H_z	5
ZK_z	6
ZR_z	7
R_z	8
WP_z	9
SP_z	10
FW_z	11
U_H_z	12
U_H_ZR_z	13
U_H_ZR_R_z	14
U_H_ZK_z	15
U_H_R_z	16
U_H_AWT_z	17
U_H_ASK_z	18
U_P_AWT_z	19
U_P_FW_z	20
U_FW_z	21
U_P_z	22
U_AWT_z	23
U_ZR_z	24
U_ZK_z	25
U_R_z	26
U_R_AWT_z	27
U_ASK_z	28
P_FW_z	29
P_FW_AWT_z	30
P_AWT_R_z	31
P_AWT_z	32
P_R_z	33
FW_AWT_z	34
AWT_R_z	35
ASK_z	36
U_H_ZK_R_z	37
U_H_t	38
U_H_ZR_t	39
U_H_ZK_t	40
U_H_R_t	41
F_m	42
FO_m	43
U_H_FW_m	44
U_FW_m	45
U_FW_ZR_m	46
U_FW_AWT_ZR_m	47
P_FW_m	48
P_FW_AWT_m	49
FW_AWT_m	50
FW_m	51
VF_f1	52
U_H_VF_f1	53
U_VF_f1	54
VF_ZR_f1	55
NF_f1	56
U_H_ASMI_tr	57
U_H_ASMG_tr	58
U_ASMI_tr	59
U_ASMG_tr	60
Trigger_Salz	61
Trigger_Fuellzaehler	62
Trigger_ResetFuellzaehler	63
Trigger_Warmwasser	64
Trigger_ACQS	65
Trigger_F1	66
Trigger_RT	67
U_ZH_t	68

_z -> dependet time
_t -> dependet temperature
_f1 -> dependet filling switch
_m -> dependet filling amount
_tr -> dependet turbidity sensor

U	->	Mainpump
P	->	Drainpump
PA	->	Break
AWT	->	Outletvalve Heatexchanger
H	->	Heater
ZK	->	Dispenser Rinse Aid
ZR	->	Dispenser Detergent
R	->	Regeneration valve
WP	->	Alternate Draining (5sec. Mainpump, 5sec. Drainpump)
SP	->	Sputter Draining (5sec. Drainpump, 5sec. Break)
FW	->	Filling Softwater
ASK	->	Calibrate Aquasensor
F	->	Filling (Complete Fillingstep, Parameters for Heatexchanger)
FO	->	Filling (Complete Fillingstep, Parameters for Waterinlet)
VF	->	Filling Heatexchanger (With Time memorising)
NF	->	Filling Heatexchanger (Without Time memorising)
ASMI	->	Aquasensor measurement infrared
ASMG	->	Aquasensor measurement green
ZH	->	Time and Temperature heating

Trigger

Salz	Check if Sald is empty
Fuellzaehler	Check if Regeneration is necessary
Reset Fuellzaehler	Set Fillingcounter to "0"
Warmwasser	Check if the Dishwasher is connect to a hotwaterinstallation
ACQS	Chck if the Aquasensor is wrong connected
F1	Check if F1 is switched
RT	Calculate Time to End

5/18/2005 ESS / Heidel

Figure 37 Notations for Customer Service Program