Aalto University
School of Science
Department of Applied Physics
Degree Programme of Engineering Physics and Mathematics

Lauri Neuvonen

Automated Testing of Solid Oxide Fuel Cell Units' Control System

Master's Thesis
Espoo, September 19, 2011

Printing date: September 21, 2011

Supervisor:	Peter Lund, Aalto University
Instructors:	Antti Pohjoranta, D.Sc., VTT
	Toni Ollikainen, M.Sc., Wärtsilä

Aalto University
School of Science
Department of Applied Physics
Degree Programme of Engineering Physics and Mathematics

ABSTRACT OF
MASTER'S THESIS

Author:	Lauri Neuvonen
Title of thesis: Automated Testing of Solid Oxide Fuel Cell Units' Control System	
Date: September 19, 2011	**Pages:** 12 + 60
Professorship: Energy Technologies	**Code:** Tfy-56
Supervisor:	Peter Lund
Instructor(s):	Antti Pohjoranta, D.Sc., VTT Toni Ollikainen, M.Sc., Wärtsilä

This thesis presents a testing protocol and a testing platform for the purpose of automatically testing the control unit of a solid oxide fuel cell system. The work was commissioned by Wärtsilä - a Finnish company involved in maritime solutions and distributed power generation.
The protocol was constructed to ensure all necessary things are taken into account during testing. The platform provides a set of tools for realizing the tests. The different parts of the platform were taken into use and configured to work together. Three case examples of test runs were performed to test the functionalities of the platform.

Keywords:	solid oxide fuel cell, automation, testing, simulation assisted testing, automated testing, programmable control logic, process control.
Language:	English

Aalto-yliopisto
Perustieteiden tiedekunta
Teknillisen fysiikan laitos
Teknillisen fysiikan ja matematiikan tutkinto-ohjelma

DIPLOMITYÖN TIIVISTELMÄ

Kirjoittaja:	Lauri Neuvonen
Otsikko: Automated Testing of Solid Oxide Fuel Cell Units' Control System	
Päivämäärä: 19. syyskuuta 2011	**Sivuja:** 12 + 60
Professuuri: Energiateknologiat	**Koodi:** Tfy-56
Valvoja:	Peter Lund
Ohjaaja(t):	Antti Pohjoranta, TkT, VTT Toni Ollikainen, DI, Wärtsilä

Tässä diplomityössä kehitettiin testausprotokolla ja -ohjelmistopaketti kiinteäoksidipolttokennon ohjausjärjestelmän automaattiseen testaamiseen. Työn tilasi Wärtsilä, joka toimii merenkulun ja hajautetun energiatuotannon aloilla.
Protokollan tarkoitus on varmistaa, että ohjausjärjestelmää testattaessa tulevat kaikki tarpeelliset asiat tehdyiksi. Ohjelmistoalusta puolestaan tarjoaa työkalut testaamiseen. Ohjelmistoalustan eri osat otettiin käyttöön ja ne asennettiin toimimaan yhdessä toistensa kanssa. Ohjelmistoalustan toimivuus testattiin suorittamalla kolme eri testitapausta.

Avainsanat:	kiinteäoksidipolttokenno, automaatio, testaus, simulaatio-avusteinen testaus, automatisoitu testaus, ohjelmoitavat ohjainlogiikat, prosessin ohjaus
Kieli:	englanti

Foreword and Acknowledgements

This thesis was written between January and July in 2011. As my studies seem to finally come to an end, I wish to thank some people who have influenced my journey.

First of all, I wish to express my deepest gratitude to my instructors, mr. Antti Pohjoranta and mr. Toni Ollikainen. Their wise input and most helpful attitude made writing this thesis fairly painless. I also wish to tank prof. Peter Lund for supervising this thesis. My parents deserve a huge thankyou for supporting me throughout my studies, I finally made it!

This part of my life wouldn't have been what it was without the Teekkari-community in Otaniemi. And it wouldn't be what it will be. Being part of it also allowed me to meet people from all over Finland and the rest of the world. So hugs, kisses and cheers to all the people I sang with, sat with, adventured with and became friends with! You mean the world to me!

Espoo July 13th 2011.

Lauri Neuvonen

Contents

List of Abbreviations

ACL Access Control List

BoP Balance of Plant system

FAT Factory Acceptance Test

FC Fuel Cell

IS Instructor's Station

OLE Object Linking and Embedding

OPC OLE for Process Control

PLC Programmable Control Logic

SOFC Solid Oxide Fuel Cell

List of Tables

List of Figures

Chapter 1

Introduction

This thesis aims to explore the subject of solid oxide fuel cell (SOFC) systems, their role as a new energy technology in solving the global sustainability questions, and especially the problems and backgrounds related to testing such systems. A computer aided step-by-step testing protocol and a testing platform for testing the programmable control logic (PLC) unit responsible for system control are presented. The vulnerabilities and development suggestions for the protocol and platform are discussed.

Given the subject and goals of this thesis work, SOFC systems, their simulation, automation and automation testing related issues are emphasized in this thesis report.

1.1 Global Demand For New Energy Technologies

The increasing global pressure to control the effects of modern human life on our planet has led many companies to develop more environmentally friendly technologies and practises. This trend has especially affected companies involved in energy production and energy intensive businesses as energy production is one of the main sources of greenhouse gas emissions and a heavy user of global non-renewable resources.

Another influential factor is the inevitable depletion of global oil resources. Even though the time line of this is debated and relies on political decisions, economics and discoveries of new oil reserves, the fact that oil will eventually run out remains. This will, along environmental issues, cause pressure to develop new energy solutions for power generation and propulsion technologies for transportation.

Electricity is a viable solution in many cases. As a highly flexible energy carrier producible by many different sustainable methods such as solar and wind power, it is the long term solution for most of mankind's energy needs. At the moment, though, the motor and battery technologies have not yet reached a level that would allow the economically viable use of electricity as the main energy carrier for very large motors, or in situations where recharging is not frequently available.

There are two main problem fields related to the aforementioned global ecologic and economic phenomena that motivate the development of fuel cell systems. These are presented below. At the moment there exists no single or simple solution to these problems - just promising new technologies requiring more development. It is probable that the "solution" will consist of several technologies used in different situations and applications according to their individual strengths.

Decentralized energy production is one of the concepts expected to play a big part in future energy production. Instead of producing energy in a large plant in one place and then transporting it, it would be more flexible and efficient to produce energy close to where its being consumed. This concept calls for new energy technologies as the conventional energy producing methods are very inefficient in small scale. Solar and wind power are two of the most publicly discussed technologies for achieving this. Their benefits include low pollution - no pollution during operation - and free, sustainable sources of energy. They do, however, have some drawbacks. One of the main obstacles to overcome is their dependency on the local weather: no sun or wind - no power. Supportive energy production methods are therefore required as a backup.

There also exist situations and applications where a stand alone energy backup system is necessary - in other words, decentralization for safety reasons. This applies to hospitals, ships, nuclear power plants etc. that can't rely solely, or at all, on electricity from a grid. In addition, they can't rely on sources of energy that are vulnerable to weather conditions as a backup. Today, most of these backup systems are based on diesel-generators or, in larger cases, on small power plants that burn gas or oil. Ships for example are at the moment bound to use energy carriers that can be transported along in conventional containers such as tanks. New solutions for energy storage are constantly being developed but they have not yet reached a level that would allow their use in large scale [16].

Transportable energy solutions, such as small scale electricity generators, can also be considered a form of decentralized energy production. Even though transportable energy production methods are not a very significant factor from the viewpoint of global economy, their local and temporal effects can be significant - in crisis situations, for example.

Transportation has traditionally relied on crude oil-based fuels as it's main source of energy. Petroleum products account for 95% of transportation's total energy consumption with the only main exceptions being railways, that use electricity - of which a large portion is produced by fossils, and biofuels [15]. This cannot be considered a sustainable situation in the long term for three main reasons: Crude oil will eventually run out, before this it's price will probably rise to unfeasible levels business wise and even if the remaining resources could be fully utilized it would cause too much pollution. Emission sanctions will also add up to the expenses of crude oil based fuels.

The need for transportation itself is growing due to globalization, the economical growth in the developing countries and population growth which leads to the inevitable need to replace crude oil as the industry's main energy source [15]. The ratio of produced oil per person has already peaked [16] implying that there's less and less oil available for transportation (as well as other) purposes per capita. The same arguments can be applied to other fossil fuels also, even though emissions and price per kilowatt vary depending on the fuel in question.

There are a few possibilities for replacing fossil fuels. Modern combustion engines are being developed to allow the use of biofuels and at the same time methods for turning waste into combustible fuels are researched. The amount of biofuel that can be sustainably produced depends on the advancement in the research of 2nd generation biofuels. This means there's at the moment no guarantee that all the needed fuel could be produced as biofuels [12].

Hydrogen can also be used in combustion engines if raised to high enough pressures. However, it can't be recovered straight from the nature but needs to be produced from, for example, natural gas that is the source of 95% of all the hydrogen produced today [16]. This requires energy and causes emissions thus reducing the efficiency of hydrogen as an economic and ecologic energy carrier.

The efficiency of any type of a combustion engine is in addition limited due to the thermodynamical limits. Therefore the problem cannot be solved solely by increasing the efficiency of these engines either. It is evident that new types of engines must be introduced and researched with emphasis on efficiency and sustainable fuel options.

1.2 Solid Oxide Fuel Cell Technology

One of the most flexible future energy conversion solutions is the solid oxide fuel cell (SOFC) technology. Its benefits include high fuel flexibility, scalability in total power, energy conversion efficiency and low CO_2 and particulate matter (PM) emissions per kilowatt.

The core component in a SOFC system is the solid oxide fuel cell, which is capable of electrochemically converting the chemical energy of a fuel and an oxidant into electricity and heat without irreversible oxidation [21]. Individual cells are stacked to achieve higher total voltage and a balance of plant system is built around the stack to ensure optimal working conditions and proper fuel and air feeds.

The SOFC's capability of converting chemical energy straight to electricity and heat enables very high efficiencies. Especially the electric efficiency of SOFCs is very high compared to conventional electricity producing methods due to the form of energy being changed only once. Heat is considered to be waste in most cases but, if available in high enough temperatures, it can be utilized as is or, for example, in a combined cycle using a steam turbine and a generator to produce additional electricity. The efficiencies reach up to 60% for electricity and 90% combined [17].

SOFC systems produce some waste gases depending on the fuel gas. Compared with combustion engines the emissions from SOFC systems are however considerably lower as the efficiency of the system is high and most of the carbon exits the system as CO_2 [14] when an afterburner is used.

Should hydrogen some day become economically feasible as a fuel, the SOFC will readily be able to use it to produce virtually non-polluting energy and fresh water, an increasingly critical global resource, as a byproduct. As both of these scenarios can be considered probable, there exists a great motivation to research SOFCs from these points of view also.

SOFC systems can theoretically be scaled up without limit by connecting units in series. In practise, however, physical size limits and the combined effect of the complexity of the manifolding of individual cells and the fact that individual cells cannot be however large, limit the scaling and therefore the maximum power of a single fuel cell system.

At the moment SOFC systems also suffer from high production and maintenance costs and unreliability. These problems will most probably be overcome in time but for now they inhibit the use of SOFC systems in large extent [2].

1.3 Wärtsilä's Fuel Cell Project

Wärtsilä has so far developed several increasingly powerful SOFC system prototypes the latest being a 50kW unit capable of producing electricity to the national electric grid. Wärtsilä has researched solid oxide fuel cells for use as auxiliary power units in ships and for purposes of decentralized power generation. The same design will be applicable in both situations, even though, depending on the situation, some external systems might be required for fuel preprocessing for example. There are many problems to be solved until commercial success can be achieved but Wärtsilä is committed to developing it's SOFC technology to that point. The following quotation is from Wärtsilä's fuel cell strategy [13]:

> Wärtsilä will generate new business opportunities and support existing product portfolio by developing and commercializing fuel cell products. Fuel cell products will expand Wärtsilä power solutions offering by providing highly efficient and clean power generation products at lower power range (< 5 MW).

Having a SOFC unit as an auxiliary power unit on a ship will eliminate the need to run the ships main engines or other internal combustion engines on very low efficiency only to create the necessary electricity for lighting and other systems during stays in harbors. This solution will reduce emissions and lower the expenses related to fuel usage and emission sanctions. The fuel gas needed by the fuel cell system can be liquified and transported along.

For decentralized power generation a pilot project of a SOFC system running on landfill gas was built for Vaasa Housing Fair in 2008. The unit achieved a 47% peak net efficiency producing both electricity and heat for the housing fair district. The unit was on a scheduled maintenance break when this thesis was written, having operated for 3338h in total [2].

To achieve commercialization Wärtsilä's SOFC systems are constantly being developed requiring frequent changes and updates to the process. This in turn requires changes to the automation system that has to be tested after every revision to ensure safe and efficient operation of the fuel cell system. This gives rise to the problem this thesis tries to solve in it's part.

The automation system has two main functions regarding SOFC system operation - safety and process control. It consists of two separate subsystems - a hardwired safety circuit mainly taking care of personnel safety and a process control system consisting of a control unit and the necessary field equipment. The control unit, implemented in programmable control logic (PLC), is used both for control and safety purposes. In this thesis the main focus is in testing the PLC unit.

The complexity of the SOFC system demands for an automation system - it is impossible to control the system even nearly optimally manually. The automation system used in Wärtsilä's SOFC unit includes a few hundred I/O points and the process control unit does some complex real time calculations based on process measurements.

To determine whether the programs within the automation system work correctly they can, and should, be *verified*. Verification can be divided into two subcategories: dynamic verification i.e. testing by experimentation and static verification i.e. analysis of the written code. The protocol developed during this thesis is meant for dynamic verification of the whole automation logic excluding the hardwired subsystem.

It is very beneficial to use a simulator-based testing platform to test the automation system. This is mainly due to the fact that failures in automation can cause severe damage to the controlled system - and in a complex system, such as the automation system of a SOFC unit, these failures are very probable during the development phase. Another great advantage is that, when using a simulator, there's no need for controlled heating, cooling or other transitional phases during the tests - one can simply load a model in the desired state. This saves time and reduces system wear due to transient induced degradation of components. Therefore, by using a simulator model instead of the actual SOFC unit in automation testing, it's possible to reduce safety risks and save resources.

The rapid development of modern computers has made it possible to use normal desktop computers in simulation tasks. Commercial process simulator softwares have also developed to a state where they allow the simulation of very complex systems such as nuclear power plants in near full scale. Simulation can therefore be considered beneficial also economically as there's no need for special computer equipment or marginal know-how related to it.

Wärtsilä has developed fuel cell models in collaboration with VTT for its latest fuel cell systems including a purpose built fuel cell stack model and models for most of the BoP components. These models, implemented using the Apros (Advanced PROcess Simulator) simulation software, simulate the fuel cell system with sufficient accuracy in real time or faster - in the case of fast transient simulations the speed can however be lowered to gain more accuracy [20]. For automation testing purposes the simulator model is connected to the physical automation unit which measures the state of the simulated process and sends control commands to the modeled process components. The automation unit therefore acts exactly as it would when connected to a real SOFC unit.

Testing the automation logic has so far been done by hand requiring a lot of active control and surveillance by development personnel keeping them away from actual development work. Much of the testing work is of such nature that it is possible to automate it. The sheer amount of work related to manual testing has also inhibited testing in full extent. There also hasn't existed a protocol to be followed during the testing adding to the probability of inadequate testing due to test planning errors.

Chapter 2

Solid Oxide Fuel Cell Systems

This chapter explains the basic working principle, components and properties of a solid oxide fuel cell system.

2.1 History of SOFCs

The first propositions for using solid oxide fuel cells as a way to generate electricity from fuel via electrochemical reactions were made by Walther Nernst and his colleagues at the end of nineteenth century. Nernst's main breakthrough was to find that the conductivity of stabilized zirconia behaved differently in different temperatures. Especially the fact that from 600°C to 1000°C zirconia behaved as an ionic conductor was a key discovery that led to the invention of fuel cells [21].

Not much happened until the 1930s when Baur and Preis demonstrated a laboratory scale fuel cell. They speculated that some day fuel cells might compete with batteries. In the 1950's a straightforward testing design was developed for stabilized zirconia discs and its still in use today. The problems related to planar stacks led to the development of tubular designs. Both designs are still used and researched further today [21].

At the moment fuel cells are becoming an increasingly beneficial technology due to their efficiency, fuel flexibility and low emissions. A lot of work still needs to be done to reduce manufacturing costs and to improve their endurance. It can be said that the over a hundred year-long path from the first ideas is at the very moment reaching the goal of becoming an everyday technology.

2.2 SOFC Working Principles

2.2.1 Reactions

The Basic Working Principle of a Fuel Cell

A fuel cell (FC) consists of an anode, a cathode and an electrolyte element. The anode and cathode, both consist of porous material capable of catalyzing the required reactions. The electrolyte is in between the anode and cathode which are additionally connected via an external circuit consisting of leads and a load or the grid. The general FC structure is shown in figure 2.1. The following working principle applies to most fuel cells, including SOFCs. The reasons why a SOFC is more flexible fuel wise are discussed in 2.2.1.

The core of the cell is the electrolyte that allows ions to pass through it but does not allow the same for electrons. The type of ions that travel through the electrolyte varies depending on the type of the fuel cell in question.

On the anode hydrogen H_2 is ionized catalytically (2.1) and reacts with the oxide ions to form water (2.2). This reaction also produces heat.

$$H_2 \rightarrow 2H^+ + 2e^- \tag{2.1}$$

$$2H^+ + O^{2-} \rightarrow H_2O \tag{2.2}$$

The electrons released on the anode travel through the load to the cathode where they ionize more oxygen. This process is shown in figure 2.1

The Working Principle of a SOFC Utilizing Internal Reforming

The reason a SOFC is so flexible fuel wise is the fact that it runs hot enough to make a platinum catalyst, easily coked by carbon, unnecessary and thus allowing the use of hydrocarbons as a fuel. In addition to this, it's also capable of internally reforming methane into hydrogen and carbon monoxide. The following process is unique to SOFCs. The

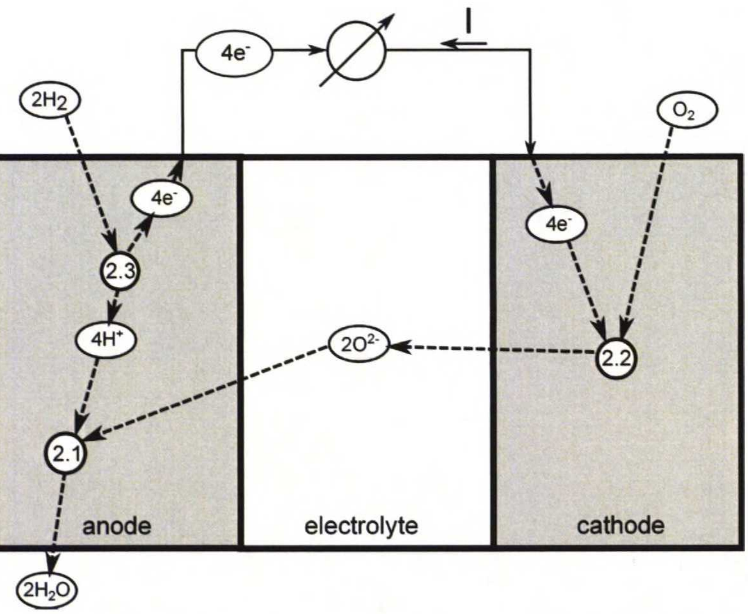

Figure 2.1: The basic reactions taking place inside a fuel cell to produce electricity (and heat). In this case, the electrolyte allows oxide ions to pass through it.

reactions discussed here are still only additions to the basic fuel cell reactions discussed before and responsible for electricity generation also in the case of a SOFC.

In the case of a SOFC, oxide ions O^{2-} travel through the electrolyte. The cathode reaction

$$O_2 + 4e^- \rightarrow 2O^{2-} \tag{2.3}$$

provides oxide ions by catalytic ionization of oxygen in the air.

Methane CH_4 is fed to the anode and it reacts with water according to the methane steam reforming reaction (2.4) where CH_4 breaks into carbon monoxide and hydrogen. Carbon monoxide reacts with steam according to the water gas shift reaction (2.5) to form carbon dioxide CO_2 and H_2. Part of the H_2 is used in the discussed reactions and the rest of it flows out from the cell. Excess H_2O and CO_2 exit the anode with the unused fuel. The chemical reactions taking place during the reforming process are [11]:

$$CH_4 + H_2O \rightarrow 3H_2 + CO \tag{2.4}$$

$$CO + H_2O \rightarrow CO_2 + H_2 \tag{2.5}$$

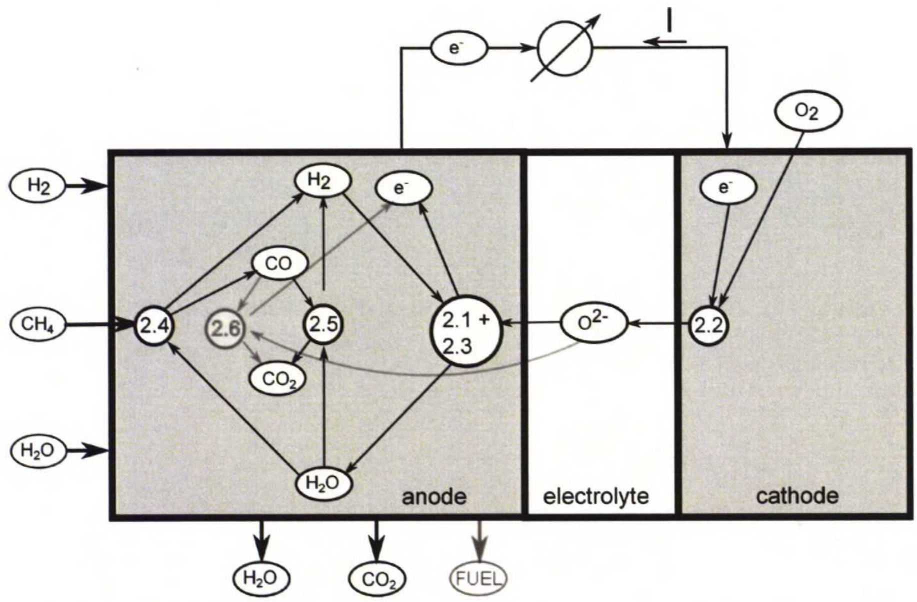

Figure 2.2: The main reactions taking place inside a solid oxide fuel cell. The figure does not count for the amounts of reactants involved in reactions.

Figure 2.2 shows how the different reactions contribute to one another. In addition to the methane used as the main fuel smaller amounts of hydrogen, water and other gases are required to maintain the thermal and chemical balances inside the stack. The figure includes only the reactions and reactants involved in the dominating and critical reactions.

It's possible that CO also oxidizes electrochemically to form more electricity according to the reaction

$$CO + O^{2-} \rightarrow CO_2 + 2e^- \tag{2.6}$$

The main electricity producing reaction is nevertheless (2.1).

In addition to water, hydrogen and carbon dioxide, unused fuel exits the anode. The anode gas is recycled so that the fuel and water it contains can be utilized in the anode reactions.

The methane reforming reaction (2.4) can take place either outside the cell in an external reformer unit and / or internally inside the cell. This affects the thermal balance of the cell stack as is described in 2.2.3. This reaction is in part behind the fuel flexibility of the SOFC as it enables the derivation of hydrogen from an abundantly available and easily stored fuel. Steam reforming is also more efficient than other methane reforming reactions even though other reactions are also possible. These are not covered in this thesis.

Different types of hydrocarbons can also be used as fuel, as long as the cell system is able to reform them into CO and H_2. This is also one of the main strengths of SOFCs compared to other fuel cell designs [21]. Fuel flexibility can be further increased by using a external reformer.

2.2.2 Oxygen vs. Carbon Balance

It's important that there's enough oxygen present on the anode. This ensures that no solid carbon can form from carbon dioxide, which could cause coking and thus make the cell dysfunctional. The abundance is ensured by feeding steam to the anode - the water gas shift reaction (2.5) becomes dominant.

Once the cell has reached its nominal operating conditions, the amount of water produced by reaction (2.2) should theoretically be sufficient to eliminate the need for an external steam supply. This is usually not achieved and thus recycling of water from the anode is required to satisfy the water consumption of the reforming reactions. If a fuel requiring no reforming is used, there's of course no need for water recycling.

2.2.3 Thermal balance

All the hydrogen produced by the cell cannot be created by the internal reforming reactions - therefore the cell needs to be fed some hydrogen gas to function properly. This is due to the fact that reaction (2.4) is very endothermic [11] and would cool the cell too much without the additional heat produced by the combination of reactions (2.1) and (2.2) using pure hydrogen i.e. without the reforming reaction (2.4). The reforming reaction is, on the other hand, very useful as a heat sink. It eliminates the need for cooling by an air stream or other methods at the same time producing hydrogen for electricity producing reactions. This has a very positive effect on the electrical efficiency of the system and is a great advantage of the SOFC design compared to other types of fuel cells.

2.3 System Components and Subsystems

The components of a SOFC power system can be divided into two sections: the fuel cell stack components ("the stack") and the balance of plant (BoP) components. The stack consists of the individual fuel cells and the immediate structure supporting them, including the gas manifolds, whereas the BoP components make up the rest of the system. The stack is the heart of the system responsible for all the reactions producing electricity and some of the heat alongside the afterburner. The BoP components aim to provide the stack with the correct fuel mixture and the right amount of oxygen, as well as handle all the necessary recycling, exhaust processing, fuel cleaning etc. The reactions described in 2.2.1 demand a lot from the BoP design. A general diagram of a fuel cell system is shown in figure 2.3.

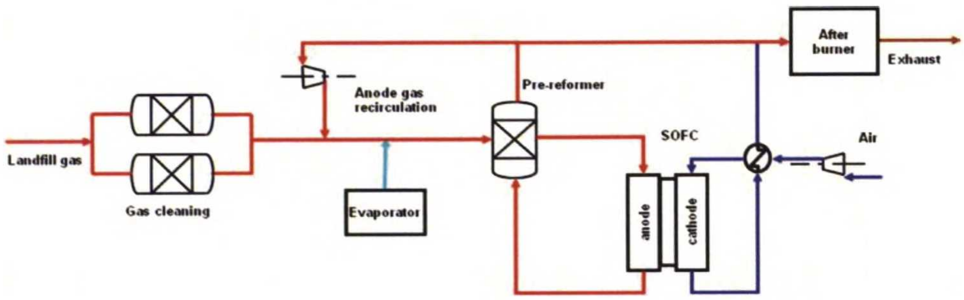

Figure 2.3: A general diagram of a SOFC system consisting of the fuel cell stack and balance of plant components. This is a diagram for a SOFC unit that uses landfill gas for fuel. The system includes a gas cleaning unit and a pre-reformer to cope with the impurities of the gas. Anode gas is recirculated to ensure enough steam is present in the reformer and also to recollect the unused fuel [13].

In addition to these process related components, the SOFC system also incorporates an automation system. Its function is to ensure the optimal and safe operation of the SOFC unit. In addition to performance optimization and ensuring safety, automation utilization has the benefit of making the work of the operator significantly easier by helping with startup and shutdown procedures, for example. It's trivially economically wiser to invest in a high quality automation system than in the additional operation personnel that would be required to replace the automation. The automation system has a significant role in ensuring that the SOFC system implements the guidelines presented in section 2.4.

2.4 Design Guidelines for SOFC Systems

2.4.1 Usability

The SOFC unit must naturally fulfill its native purpose of power production. The usability of the system in a variety of situations is, however, a more complex issue. The power output range, physical size, tolerance of different ambient conditions, fuel flexibility and such factors must all be considered and optimized to create an economically and situationally optimal system.

After on-demand usability of the unit is achieved, it must be ensured that it is safe to operate and that its operation is optimal (not to be confused with the optimal design of the system). Automation has a very integral part in meeting these goals.

2.4.2 Safety

There are several factors to be considered regarding personnel, environmental and hardware safety. The main threat posing elements are:

- Electricity
- Flammable and toxic gases
- Large pressure differences
- High temperatures

Each of these is potentially lethal for persons and harmful for the environment exposed to them. It's also noteworthy that these elements are interdependent which increases the average threat created by a single malfunction. For example, should too high a pressure difference cause an explosion, it's probable that flammable and hot gases are also released, possibly increasing the damage created by the explosion.

Hardware safety is important both from the viewpoint of personal safety and from the viewpoint of operational efficiency of the system. All hardware damage results in both wasted time and increased cost per kWh. In the worst case, damaged hardware causes effects that result in the loss of life through some of the aforementioned elements.

The safe operation of the system demands that all process values stay within limits. This is achieved by system design and process control. System design is not covered in large extent in this thesis. It must be noted however that system design is the first and the most defining factor in system safety - critical faults in system design cannot be covered by means of system control. System design can, however, make up for faults in system control mechanisms.

This said, system control is the main basis for automation design, and therefore, for automation testing also. The automation system must, at all times, ensure that critical process value limits are not crossed. Should this kind of a situation occur, the automation system must be able to perform an emergency shut down as a "last resort". Because an emergency shutdown is a violent procedure it will most likely cause some loss of life time for the most sensible parts in the system. Therefore it is to be considered a means for minimizing damage in critical situations rather than a solution for every deviation from the optimal operating conditions. Dynamic on-line control of the process is therefore essential.

2.4.3 Performance Optimization

Performance optimization aims to direct the system to a state where its output, electricity in this case, efficiency and lifetime are optimized. The optimization task can be divided into subtasks as follows:

- Maximizing (or optimizing, if maximum is non-optimal) the average power output of the system.
- Minimizing the fuel consumption per produced kWh of the system.
- Minimizing variations in the output i.e. stabilizing the output.
- Minimizing the number of violent transients in the system.
- Achieving fast and accurate control response over the system.

- Other additional tasks.

The aforementioned goals can be achieved by means of good automation design and operation. Of course, in simple cases manual control can be an option but in systems as complex as a SOFC system, utilizing automation is the only realistic way to achieve optimal performance. It's insufficient to focus on how automated controls affect the system - the way the automation itself responds and works must also be researched, developed and thereby optimized.

Different situations imply different optimal states. System and automation design must therefore always take the intended operation conditions and goals into consideration at the start of the design process. As with safety, system design flaws cannot be generally neutralized even with the best control design.

Chapter 3

The Testing Protocol

This chapter explains the developed testing protocol and its phases. The purpose of the protocol is to ensure that all necessary testing measures are taken during the automation testing process.

Automation system testing aims to ensure that the automation system functions properly and therefore controls the system in a way that satisfies the safety and performance optimization tasks described in section 1.3 and in more detail in chapter 2.

Different stages of automation testing are described in [18]. The protocol and methods described in this thesis are intended for use especially in the Factory Acceptance Testing (FAT) phase and as development tools during the implementation phase before FAT as these are the phases requiring most iterations and repeated test runs.

Factory Acceptance Tests are a set of tests that aim to make sure the unit is in working condition and functions as it should. They are performed when the unit is leaving the factory - hence the name. The unit can be considered ready after it has passed the FAT tests.

3.1 General Outline of the Protocol

The testing protocol is a collection of tools and actions organized into a sequence that aims to ensure that all the necessary requirements are fulfilled when testing a system. The requirements are derived from international and corporate standards and directives as well as from the knowledge of how the system and its individual parts function. The steps of the protocol are:

1. Collect all necessary background information about the system.
2. Collect all requirements and build the FAT list (if it doesn't exist already)
3. Examine the verbal test descriptions from the Factory Acceptance Test list.
4. Update the test database
5. Setup the testing platform as described in chapter 4.
6. Run tests.
7. Analyze test results.
8. Report test results.
9. Accept tests or modify the system and iterate tests until the results are acceptable.

These are also depicted in figure 3.1. Should a proper FAT list, i.e. built according to standards and company protocols, not exist, it would be necessary to create such a list.

The testing itself is conducted using a testing platform developed as a part of this thesis work as explained in chapter 4.

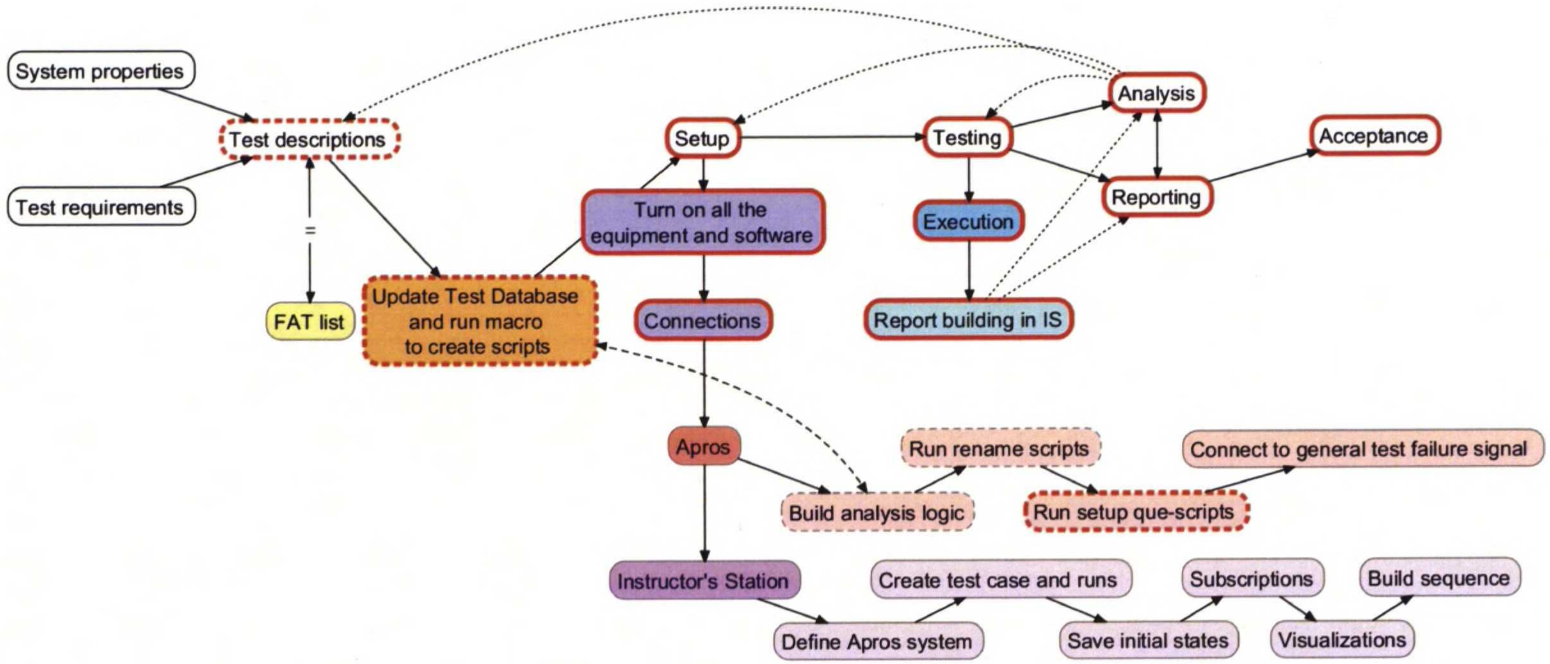

Figure 3.1: Outline of the testing protocol. The actions required every time testing is done are marked with a thick red border. Actions that have to be considered every time are marked with a dashed red border. Actions required only seldom are marked with a dashed thin border.

3.2 Standards and Test Requirements

3.2.1 Derivation of Test Requirements

Standards and codes

There exist no international SOFC specific standards yet as the technology is just starting to reach commercial use. However, many standards related to fuel cells in general exist. These have been developed by organizations such as the International Electrotechnical Commission (IEC), International Organization for Standardization (ISO), European Union and different nations. A comprehensive list of different fuel cell related standards can be found on the internet site *www.fuelcellstandards.com* [10] which also collects the relevant news and events related to the topic.

In addition, the balance of plant (BoP) system can be considered a fairly general process that involves high temperatures, pressure differences, electricity and possibly toxic and flammable gases. Therefore standards dealing with these kinds of processes should be applied to the largest extent possible.

Laws regarding, for example, personnel and environmental safety must naturally be followed. Wärtsilä's experts continually follow the different codes, regulations and standards related to the technology, it's applications and single components and relay the information to the relevant staff members.

Wärtsilä's Corporate Knowledge and Practices

The main source of test requirements is Wärtsilä's corporate knowledge of their SOFC system and how it should operate - their fuel cell team in other words. The knowledge of the system's physical constraints is transformed into limit values for different process and automation variables and these values are in turn used in the analysis logic built into the testing platform and described in detail in chapter 4.4.

The tests that need to be performed are defined by the staff member who is most familiar with the part of the system that is to be tested. This naturally requires proactive measures by the expert in question.

The tests are built according to the needs of personnel and equipment safety.

As this is written, the information regarding the test details such as limit values and time windows is stored in the control logic implementation. In other words, there exists no list or other document of this data but the information is stored in the code. During this Thesis work a test database was taken into use. It's functionalities are explained in 3.2.2.

The requirements for control of single process components are derived from the components themselves. Requirements rising from different regulations and applications have been accounted for already during the component selection process. Single components and subsystems are tested aside from control system testing.

3.2.2 The Test Database

As a part of this thesis work a database was created for the requirements as a MS Excel workbook. The database's function is twofold.

1. It demands that all the relevant information is up to date and defined in the first place for a test to be possible. This way the database directs the test definitions into a more rigorous form thus eliminating inaccuracies. No need to say, the database must be properly taken care of by taking backup copies and updating it.

2. It's a useful tool for eliminating a lot of handwork in writing the necessary scripts as it is capable of transforming the data it contains into scripts that are executable in Apros. These scripts can be used to assign correct values to the test templates that are the core Apros components in test analysis. The test templates and the use of the value assignment scripts is explained in detail in sections 4.4.1 and 4.4.2.

Screen captures of the database and the macro user interface are depicted in figures 3.2 and 3.3, respectively.

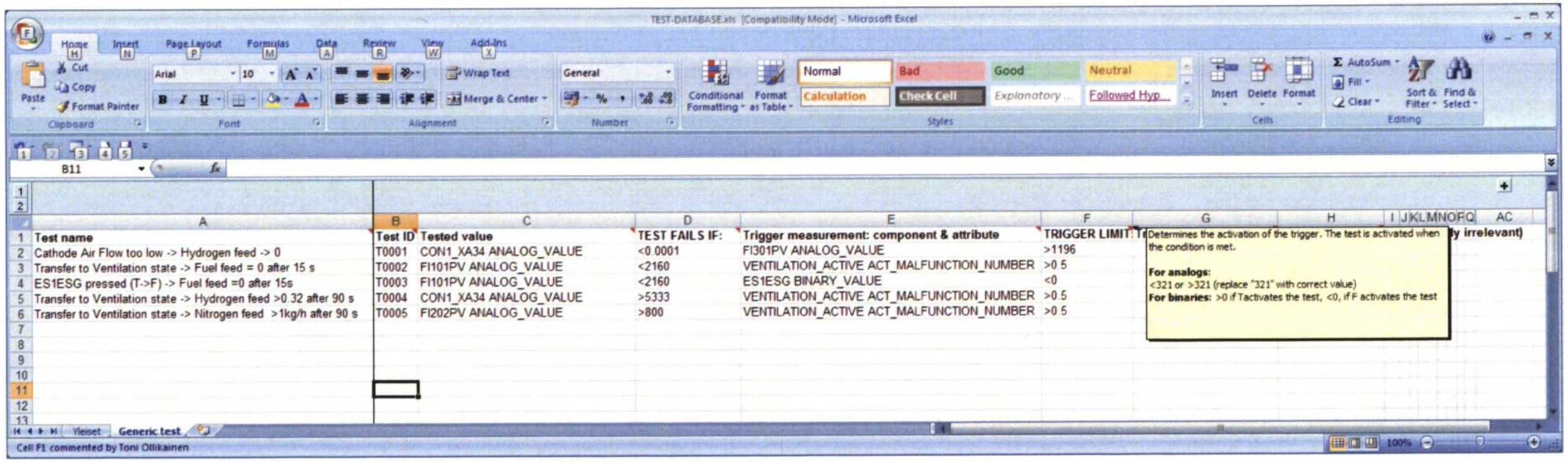

Figure 3.2: Screen capture of the test database MS Excel workbook created for test data storage and script creation. The user inserts the test related data into the cells and creates the que-scripts using the controls on the instructions and controls worksheet shown in figure3.3. The scripts will include the necessary commands for transferring the data from this worksheet into the Apros test templates.

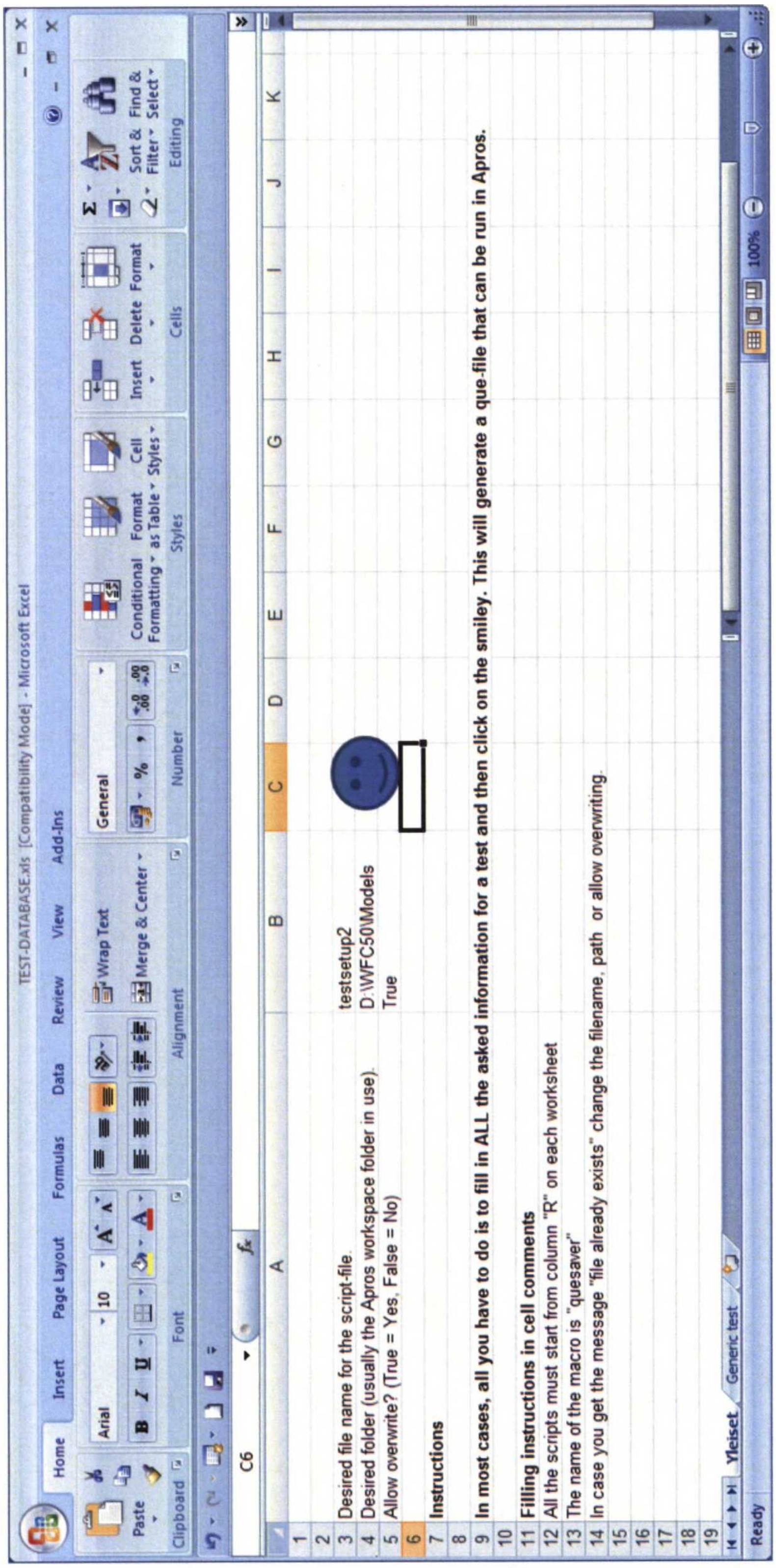

Figure 3.3: Screen capture of the instructions and controls worksheet in the test database MS Excel workbook. The macro included in the workbook saves the scripts under the name and into the folder the user defines in the filename and path cells.

3.3 The Testing Platform and Testing

The testing platform is a collection of applications used for testing the PLC control unit. Its different parts simulate the fuel cell system, provide control, analysis and results presentation functionality for testing and transfer information between parts of the platform. The platform has been built to be as generic as possible to minimize the need to rewrite or reconfigure the different parts of the platform after changes in the fuel cell system, the PLC unit or the parts of the testing platform. The platform, its components and its use are explained in detail in chapter 4.

The components of the testing platform are depicted in figure 4.5. The main components and their roles are:

Apros : Process simulation of the SOFC system, analysis of signals from PLC unit, communication between Instructor's Station and PLC unit.

Instructor's Station (IS) : Automated control of test runs, results reporting.

Database Excel workbook : Storage of test definitions and script creation for assigning values for Apros test templates.

PLC : The tested real automation and control unit.

The testing procedure consists of three steps: the setup, the test itself and report building. Starting from scratch, the setup is by far the most demanding step requiring research, planning and setting up the platform applications. After the setup is done, the testing itself requires only starting the test, supervision and iterations depending on results. Reporting is also made very easy: Instructor's Station's "report"-function produces plots of all signals that have been subscribed to. After the initial test setup the following setups can utilize most of the work done in the first setup and therefore the required amount of work is greatly reduced.

After the programs have been set up and the necessary configurations made, all that has to be done is to start the simulation. In the case of a single run, this is done by activating the run and clicking the "play"-button on IS display. In the case of a multi-run test case, the test case is ran by choosing "automatic testing" from the case's context menu. After this, all the case's runs will be run automatically by the IS software and the results are saved according to the subscriptions and displayed in the visualizations defined by the user beforehand.

3.4 Analysis

Analysis aims to either conclude, that the PLC system works correctly or, in case it doesn't, to identify the changes that need to be made. It is based on the failure signals and knowledge of how the Apros model and PLC system function and interact.

If no failure was detected, assuming the test was set up properly, the PLC unit functions as it should. If the general failure signal is positive, one then proceeds to examine the behavior of the individual failure signals. It's of course possible that more than one individual test has failed.

After identifying the failed tests understanding of the systems is used to deduce what failed and why. The behavior of the test failure signal might give some clues to this: If, for example, the failure signal has been positive only for a period of time, it's possible that the PLC system has functioned in an otherwise proper way but too slowly. One can also compare the results of different test runs to rule out or to identify possible causes for failure.

If this approach does not yield results, it's possible to build or take into use additional, problem specific analysis logics and tools to help with the analysis. One can, for example, plot the behavior of different process variables and use these to help with deduction and the identification of the source of failure.

All in all the analysis requires the use of deductive methods. The complexity of the system in practice rules out the possibility of building an all encompassing analysis logic. It would be possible to build such a logic for a finished i.e. non-development-state system by duplicating the PLC logic but, as both the fuel cell model and the PLC system are in a state of constant change due to being developed, the workload required to keep the duplicate up to date would become overwhelming. Therefore it's easier and more efficient to use deduction and a lighter analysis logic of the type presented in this thesis.

3.5 Reporting the Analyzed Results

The main tool in reporting is the FAT test document that is to be filled and accepted for the SOFC unit to officially pass the test. The FAT documentation is inspected and accepted by the person responsible for defining the performed tests or an expert with the same level of knowledge over the subject.

For more detailed documentation, additional documents can be created by using IS's reporting tool and additional visualization and documentation softwares (word processing software etc.). Wärtsilä wishes to include the information provided in these test reports in the test documentation.

At the moment there is no need for a more rigid documentation process for the test results. The FAT document is capable of transmitting the information required to accept the tests for the SOFC unit in question and, for all other purposes imaginable at the moment, tailored-to-need test result documentation is sufficient.

Chapter 4

The Testing Platform

This chapter describes in detail the different parts and functions of the testing platform, used as an integral part of the developed testing protocol, and the different procedures related to its use.

The main parts of the testing platform, as described in 3.3, are the process simulator (Apros), the test database (MS Excel workbook) and the test control software (IS).

4.1 Apros

Apros (Advanced Process Simulator) is a modeling and simulation software developed by Fortum and VTT originally to be used as a power plant simulation software [9, 18]. It has been successfully used to simulate, for example, nuclear power plants in near full scale. Apros allows the building of complicated process models by using configurable process and automation component "modules" - submodels corresponding to a single process or an automation component or auxiliary constructions used for example to connect Apros to external models. It's main function in the testing platform is to replace the physical fuel cell system thus providing a much lighter and safer testing setup compared to the situation where testing would be performed using a real fuel cell system.

An Apros model consists of interconnected modules. The model can be, and in this case is, divided into parts called "nets". These nets can also be grouped together in net groups for organizing purposes. The model also holds information of the state of the process. The model is also called a full snapshot'. A "snapshot" holds only the information of the state of the process, and not the connections between modules for example, and requires the process model in addition to itself to be useful.

For model loading purposes it's required that all the necessary initial conditions for test runs have been saved as full snapshots i.e. models in Apros' workspace folder. In case a model is altered, these initial conditions must be reproduced for that model. This can also be automated using IS runs and que-scripts but this is not discussed more thoroughly in this thesis.

Apros can be controlled in three different ways:

- Through Grades - a graphical user interface developed solely for Apros by the company Process Vision [8].
- By issuing commands through the Apros command prompt.
- Through Instructor's Station.

Figures 4.1, 4.2, 4.3 and 4.4 show screen shots of these three different controlling possibilities.

Apros also handles so called que-scripts that consist of command window commands, each on it's own line in a text file named with a '.que'-extension. These scripts can be used through the Apros command prompt and the Instructor's Station's "Script" sequence blocks.

4.2 Instructor's Station

Instructor's Station (IS) (formerly known as Testing Station) is a simulation control software developed by Fortum and VTT especially to be used with Apros. The software is still being developed and has some unfinished functionalities but it has already been used in automation system testing for Loviisa Nuclear Power Plant by Fortum [9].

IS allows the definition, automated execution and reporting of user defined test runs for Apros models. It's main function in this testing platform is to provide means for automated testing of multiple different cases in a row. IS also provides tools for reporting the test results.

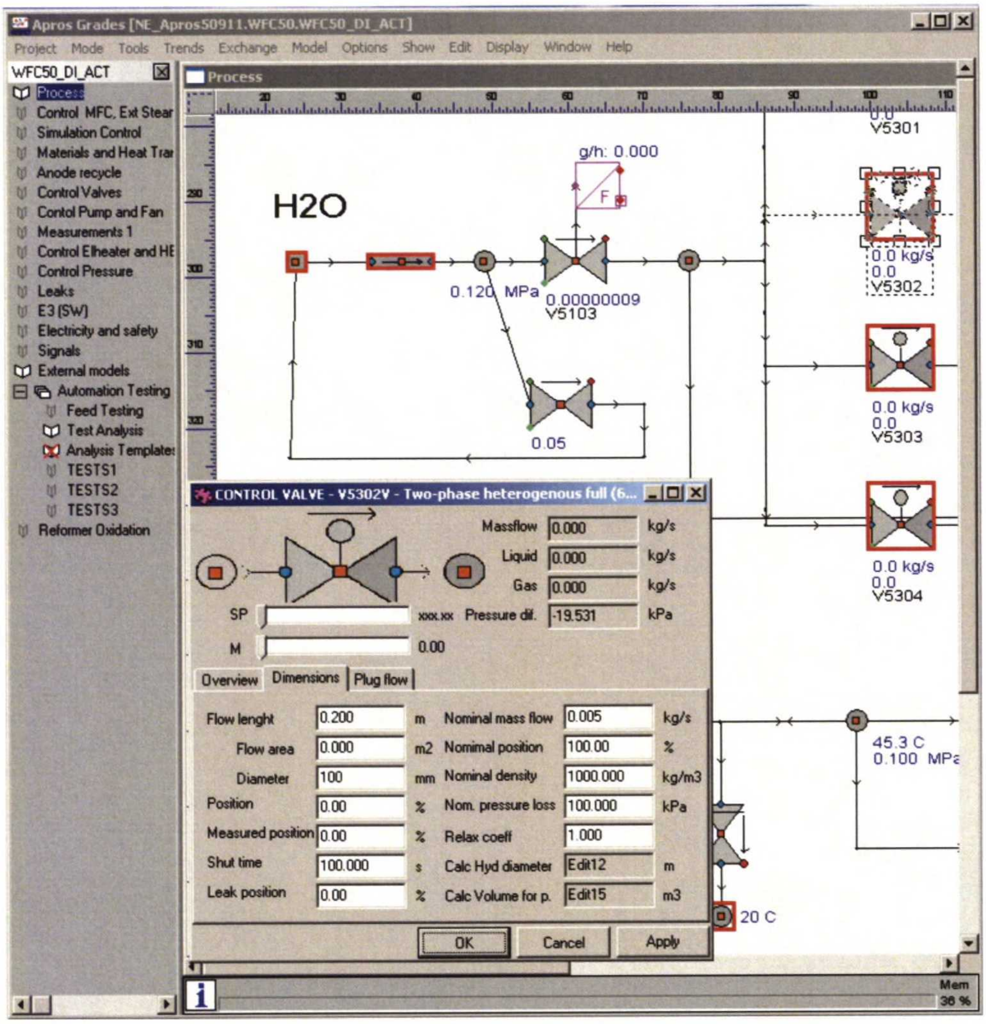

Figure 4.1: A screen capture of Grades, a graphical interface for Apros. A variety of modules and the connections between them are visible.

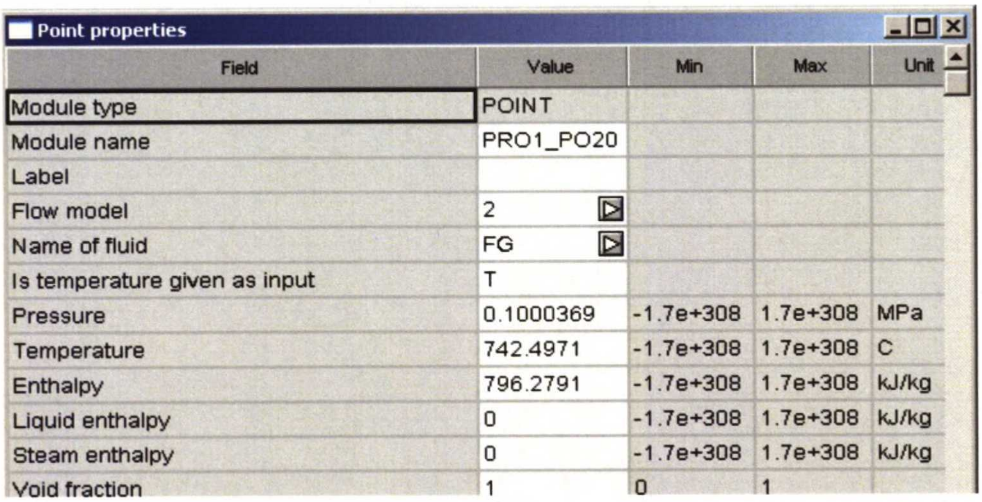

Figure 4.2: A screen capture of Grades' module editing window. Grades allows the user to change simulation parameters "on the fly".

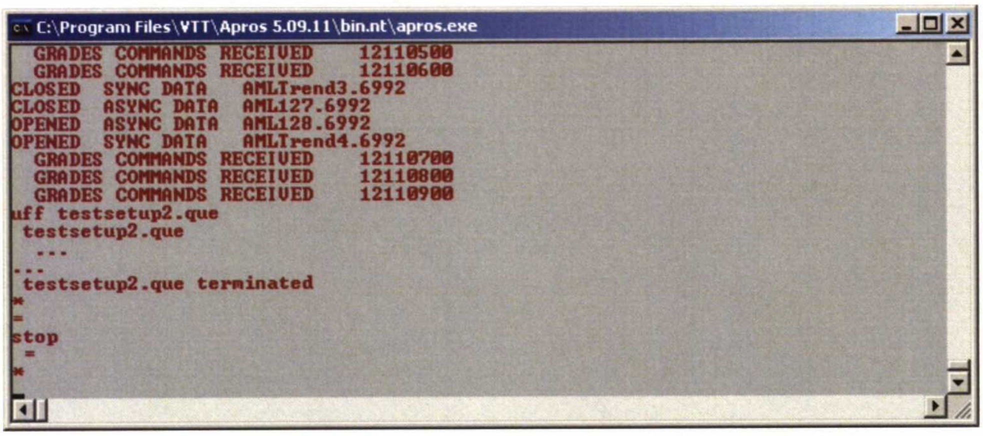

Figure 4.3: A screen capture of Apros command prompt used for giving commands to Apros in text format.

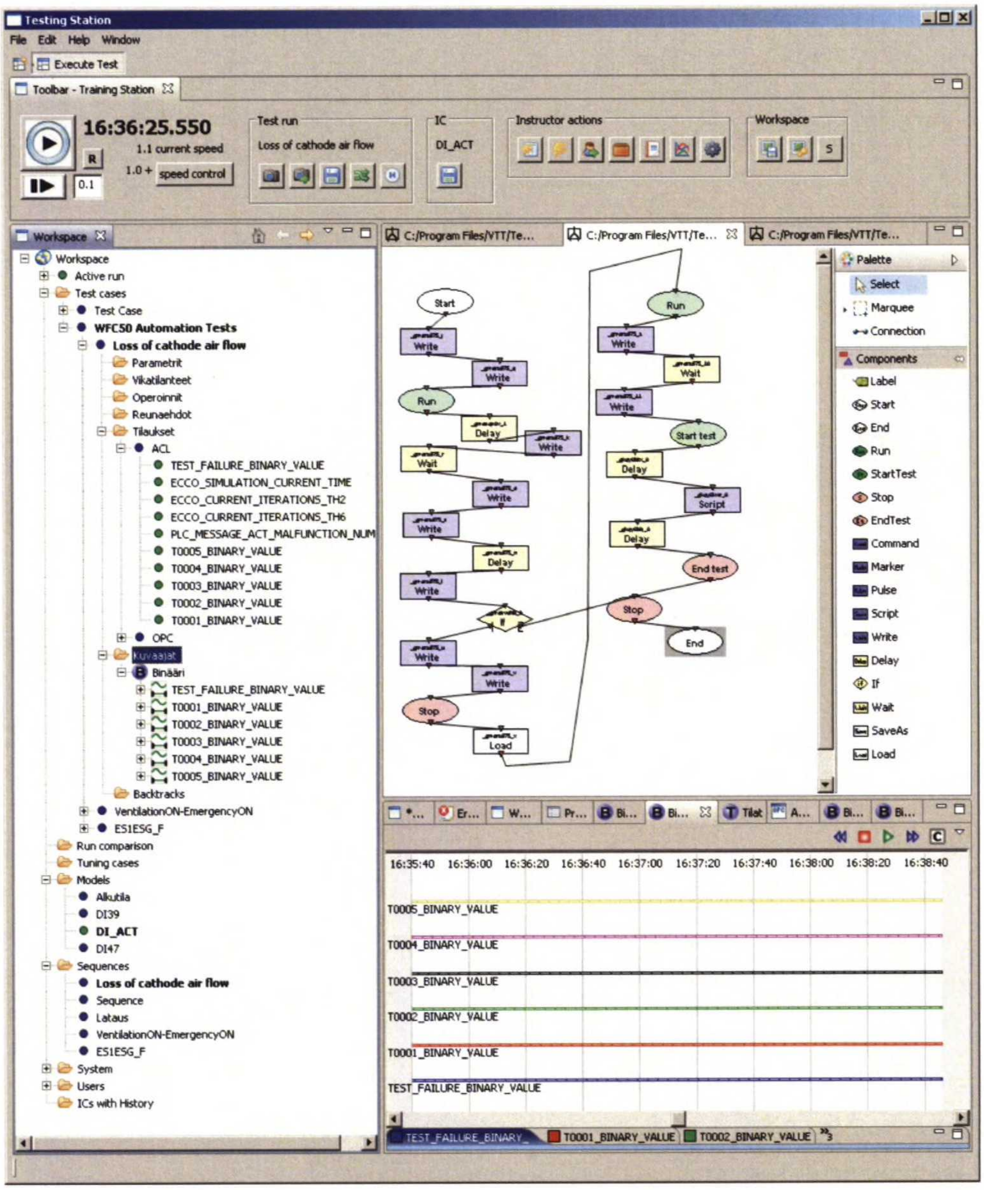

Figure 4.4: A screen capture of Instructor's Station showing the control toolbar, workspace tree, sequence editor and a visualization of six binary signals.

These tools are still unfinished and therefore provide only basic functionalities such as trend plotting and collecting the trends into a simple report sheet.

4.3 Connections

Apros and the PLC unit both communicate with other programs using an OPC server of their own. The OPC servers transferring data between Apros and the PLC unit are connected to each other via a program named X-Connector. IS, on the other hand, is connected directly to Apros' OPC server. These connection programs have no other function but to transfer data between Apros, IS and the PLC unit. Apros and IS can also be connected using an access control list (ACL) server.

Apros is used to communicate the needed commands from IS to the PLC unit and also to deliver information about the PLC's state to IS. The commands transferred to IS include only the basic commands required to run, and stop the simulation and to load and save the state corresponding with the process state of the Apros model. These commands are first written to analog signal modules in Apros using a "write" block in IS and then transferred to the Apros modules transferring data to the OPC server that in turn transfers them to the PLC unit via the X-connector and the PLC unit's own OPC server. The data transfer from the PLC unit to Apros works by the same steps but in the opposite direction.

Figure 4.5 shows how the PLC unit and IS are not directly connected to each other but Apros is used to transfer commands to, and control signals from, PLC.

4.4 Test setup

The basis for all the following steps are the verbal test descriptions found in the Factory Acceptance Test (FAT) documentation and the test requirement database introduced in section 3.2.2. A PI-chart, a system description and knowledge of the SOFC system are also vital.

The goal of the test setup is to ensure that the requirements described in the verbal test description are met, the testing itself requires as

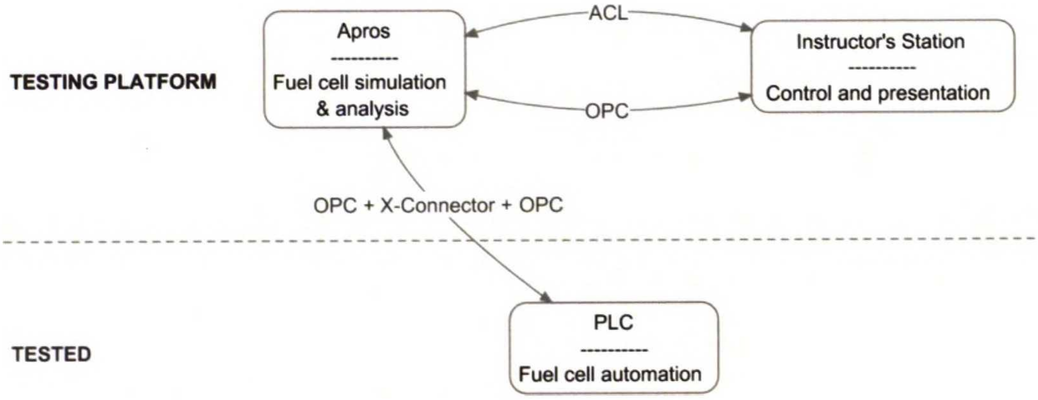

Figure 4.5: The testing setup used in testing PLC's functionalities.

little active supervision and control as possible and that the results from different tests are comparable and repeatable.

Using the information available, the person setting up the test will next find the relevant process components from the PI-chart and system description and thereafter from the Apros model. These include the

- PLC-produced control signals,
- process attributes,
- signals that PLC calculations are based on and
- signals that PLC functions are a reaction to.

Next, all the tolerance limits, limit values and delays related to the test are searched or deduced from the documentation and information of how the Apros model functions. Here "tolerance limits" mean the amount a value of a signal produced by the PLC unit is allowed to differ from the value the PLC unit is supposed to produce for the test to still be accepted. Delays define the time window in which PLC must produce the correct output, i.e. what amount of time it has to perform the necessary calculations and / or, if the control value behaves as a slope, how long the system has time to achieve the final value and stay there. The construction of the test database worksheet is shown in figure 3.2.

The collected information is inserted into the database. The following information is required:

Test description : a verbal description that describes what the test is about. A copy or a shortened version of the FAT description.

Test ID : a running number of the format "TXXXX", where the XXXX stands for a four-digit number with leading zeros filling the excess slots in front of the actual number. A test could be given the "name" T0101, for example.

Tested variable The observed Apros variable, defined by module and attribute names, that the test is based on - if this variable exceeds a certain limit while the trigger is active, the test is failed.

Test failing condition The limit value and failing side for the tested variable. Possible tolerances related to the limit value must be included in the inserted value.

Triggering variable The Apros variable that triggers the test, i.e. activates it.

Trigger's activation condition The limit value and trigger activating side for the triggering variable.

Trigger activation delay A delay value in seconds for the activation of the test from activation of the triggering variable.

Trigger deactivation delay A delay value in seconds for the deactivation of the test from the deactivation of the triggering variable. This is in most cases 0.

4.4.1 Excel Macros for Que-script Creation

The test database Excel workbook (shown in figures 3.2 and 3.3) contains a macro that can be used to create the Apros commands needed for assigning the correct values to Apros test templates. The database actually creates the commands by simple Excel functions such as IF and "&" into empty cells in the worksheet. The macro simply copies the content of these cells and saves them on separate lines into a que-script file (text file which is given the file extension ".que"). The name of the file and the folder into which the file is saved are user defined.

In most cases it's practical to save the files into the Apros workspace folder as this way they are readily usable.

4.4.2 Test Setup in Apros

The process model itself does not need modifications and it's upkeep is not a part of this testing protocol. Modifications to the analysis logics implemented in Apros might, however, need modifications as explained below. In case a new process model is created, the tests can be created by exporting the test nets of a model containing them and importing them to the new model using Apros' built in export and import functionalities. They will naturally need to be updated with the test descriptions corresponding to the new model.

Apros also doubles as an analysis software. Analysis logics built from Apros automation modules are used to analyze the control signals produced by the PLC i.e. to determine whether the PLC functions correctly. Different automation functionalities have been tested in different situations either directly by comparing the signal produced by the PLC to accepted values, or indirectly by comparing the behavior of the fuel cell system, i.e. system values, to pre-determined accepted values.

Apros Test Templates

A hundred templates were created for testing purposes. The template is presented in figure 4.6. It consists of Apros automation modules and has 8 input points and one output point. The only output is a binary signal signaling whether the test has failed (true) or not (false). This signal is connected to the "OR" module array in the "test analysis" net in the Apros model. When connected, a "true" value in this signal will be transmitted to the "test_failure" signal in the other end of the array.

The template functions according to the flow chart shown in figure 4.7. The inputs are used to determine what process attributes the logic observes, both as test triggers and the tested variables themselves, the limit values and which side of the limit values are considered to activate the test and cause its failure. The inputs also define the delays related to the test i.e. the time windows during which the process must reach an acceptable state counting from the activation of the trigger.

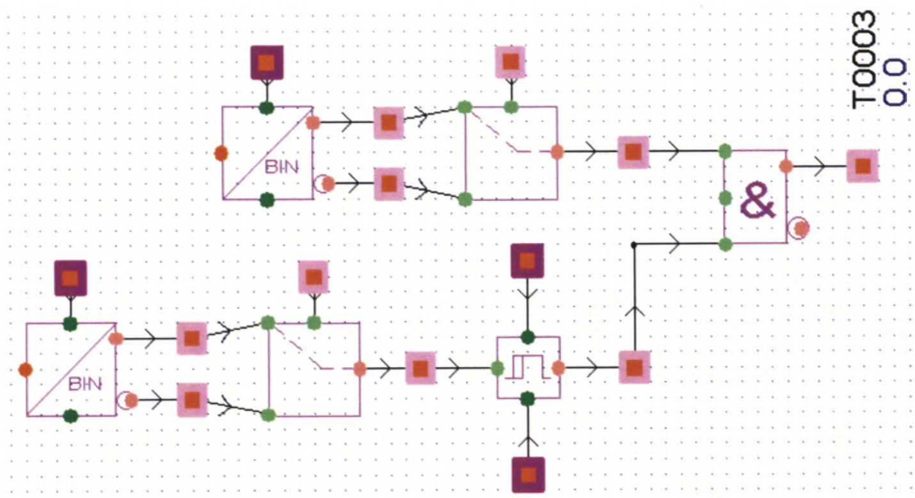

Figure 4.6: The basic test template in Apros.

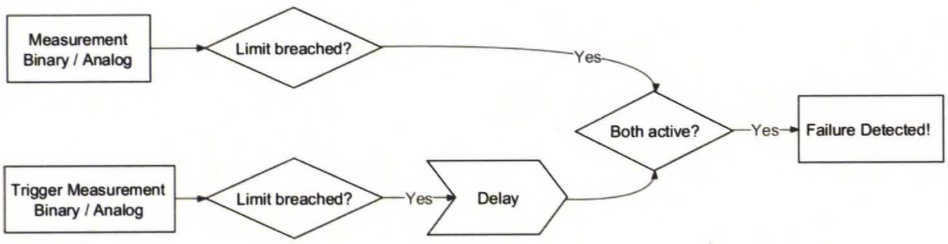

Figure 4.7: How the test template in Apros functions.

It's also possible to observe whether the process stays in a certain state long enough after the deactivation of the trigger though this is a rare and, during this thesis work, an unseen need.

The modules are assigned correct values with the help of the test requirements database. One simply enters all the information asked for in the excel-sheet and runs the macro. The macro then creates a que-script readily usable in Apros. When one runs the script, it assigns the entered values to the corresponding Apros components finishing the Apros part of the setup.

Tests Not Implementable With Readily Available Templates

There are cases where the template cannot provide the necessary functionality for testing a feature in the automation system. These include complicated situations such as non-linear or black box dependencies between variables and situations where the triggering phenomena or the tested response cannot be extracted from the process model in the form of a binary or an analog signal. It must be noted that the template does not at the moment allow for combining triggering signals so, that they both need to be activated for the test to become active.

In such cases there are a few options for performing the test. First, one can always observe the behavior of the system directly, i.e. without the aid of any analysis logics. A second option is to use an existing template but to alter it slightly, or to create a new triggering signal by deriving it from the already existing ones. Thirdly, there's the option of creating completely new analysis logics to calculate the test result. This is even advisable in situations where it's evident that there will be demand for performing other tests of the same format. In this case it would be wise to create a new template format and to modify the database Excel workbook accordingly so that the value assignment scripts for it can also be automatically created.

In any case it's very important to use the test database to document all the test cases, be they even cases tested by direct observation. The existing test template is also flexible enough to be useful in most situations. The creation of additional template forms should be considered thoroughly in order to avoid confusion and upkeep difficulties.

4.4.3 Test Setup in Instructor's Station

The Instructor's Station (IS) part of the setup consists of the following steps:

1. Defining the Apros system.
2. Creating the test case and runs.
3. Saving Apros models as initial states in IS.
4. Subscribing to the needed Apros variables for test control.
5. Creating visualizations for desired variables.

6. Building the test sequence that defines what happens during the test.

System Setup in IS

Instructors Station first needs to be told how it can connect to Apros. This is done by configuring an "Apros system" in IS. Figure 4.8 presents a screen shot of the system configuration interface. The correct values can be obtained from the Apros command prompt - it prints them when an Apros workspace is loaded.

Run Setup in IS

The tests are created and organized in IS as test cases and runs. One test case can include as many runs as is desired. A single test case can be automatically tested, meaning that IS executes all the runs included in that particular test case. One could, for example, test the automation of a fuel cell system on one shot by defining a test case named "WFC50 Automation Testing" and including all the tests included in the fuel cells FAT list as runs. A run consists of the following definitions and objects:

Initial State is the state from which the run starts. It is defined by saving an Apros model in a desired state into the IS's database.

Subscriptions define what module parameters are readable and controllable by IS. All the Apros module parameters used in the sequences must be subscribed to. Also, all the data that are to be used in creating IS visualizations (see below) must be subscribed to.

Sequence defines what happens during the run, and when it happens. The sequence is defined using the built-in sequence editor by connecting pre-defined configurable "sequence blocks" in a series. The blocks allow setting different delays, writing values for Apros module parameters and executing scripts in the Apros environment.

Visualizations are plots of user defined Apros module parameters in relation to time.

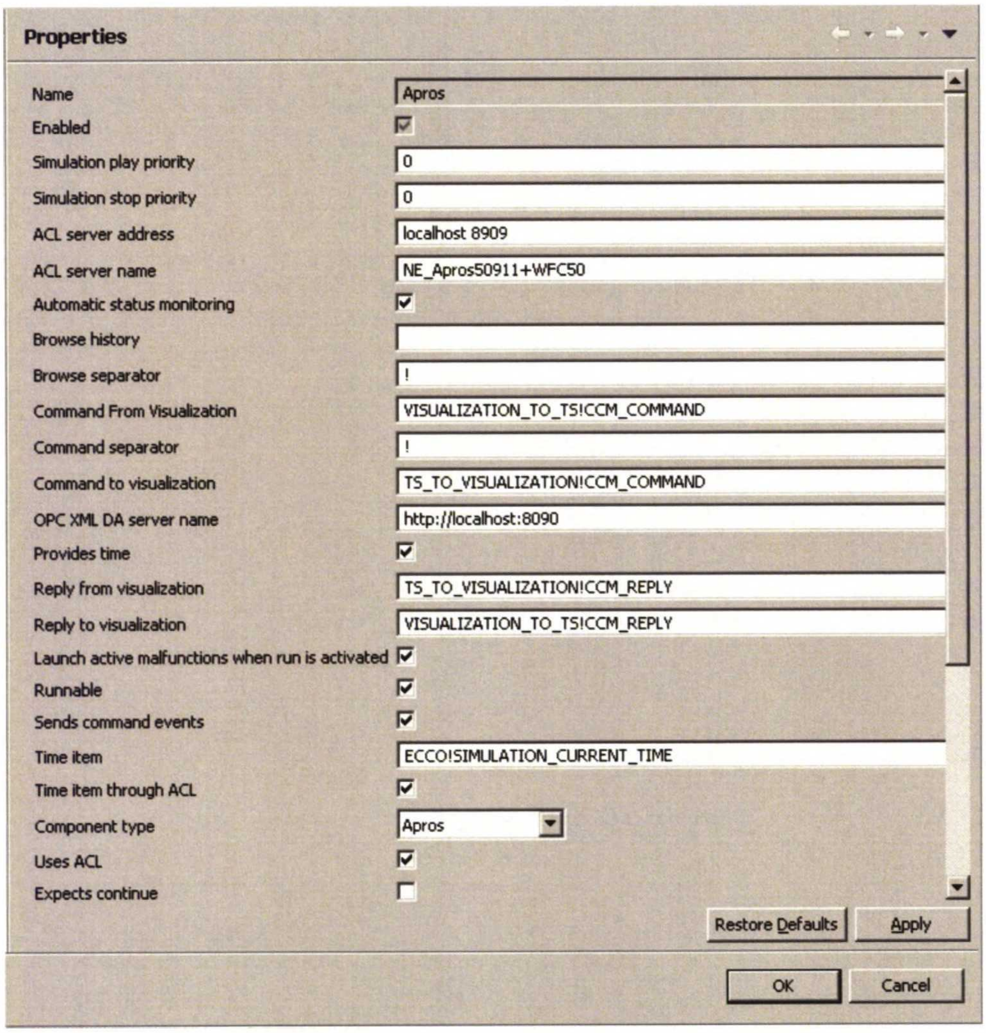

Figure 4.8: A screen capture of the dialogue window used to configure the connection between Apros and Instructor's Station.

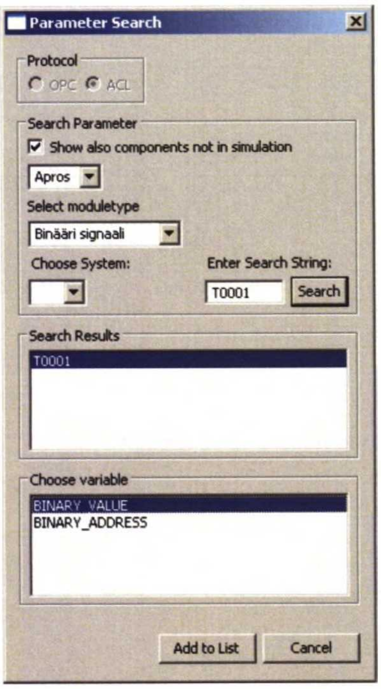

Figure 4.9: A screen capture of the subscription dialogue window.

Each run requires an initial state, a sequence and the desired subscriptions. All of these can be used in as many runs as is necessary. Next the users defines the necessary subscriptions. Subscriptions are used to transfer data from Apros to IS and also to command Apros via IS. ACL subscriptions should be used for the former and OPC subscriptions for the latter.

After the subscriptions are made, it's possible to use them to create plots of desired variables. In this case at least the test failure signals are to be plotted for test reporting and analysis. There are a few options from which the one corresponding to the type of the variable plotted should be used - "2D Trend" for an analog signal and "Binary" for a binary signal, for example. Examples of these are shown in figures 4.10 and 4.11, respectively.

How the run progresses is defined by building a sequence and assigning it to the run. Sequences are built using a graphical editor in IS 4.12. A sequence consists of basically two parts: The first part prepares the PLC unit for the test by commanding it to load the correct initial state and then starting the unit.

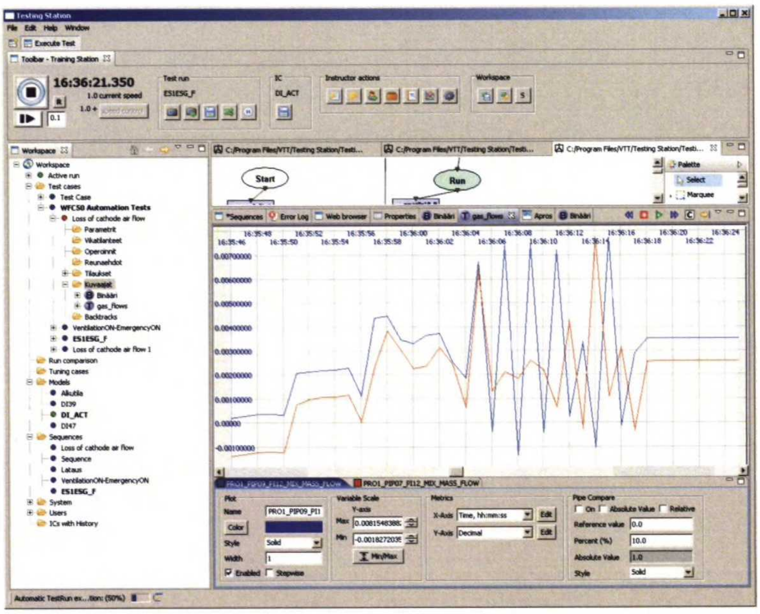

Figure 4.10: A screen capture of a 2D-trend in IS.

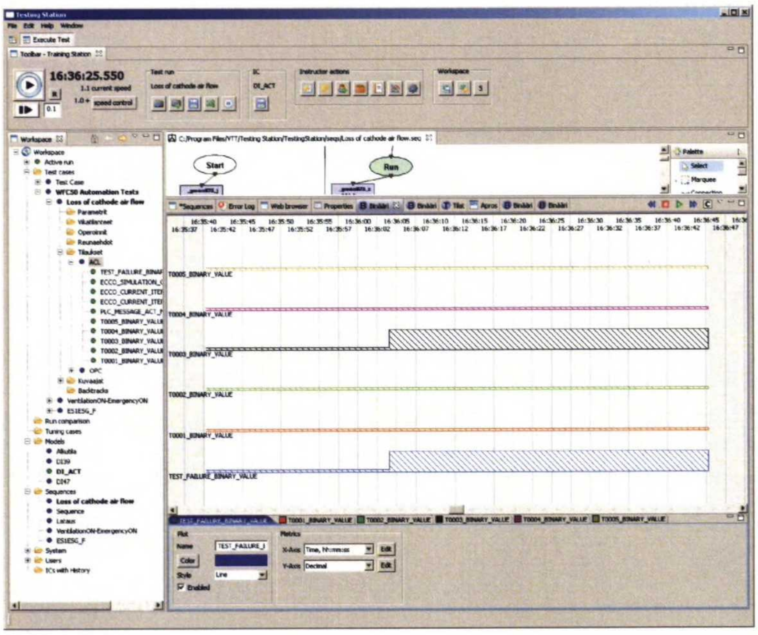

Figure 4.11: A screen capture of a binary visualization in IS.

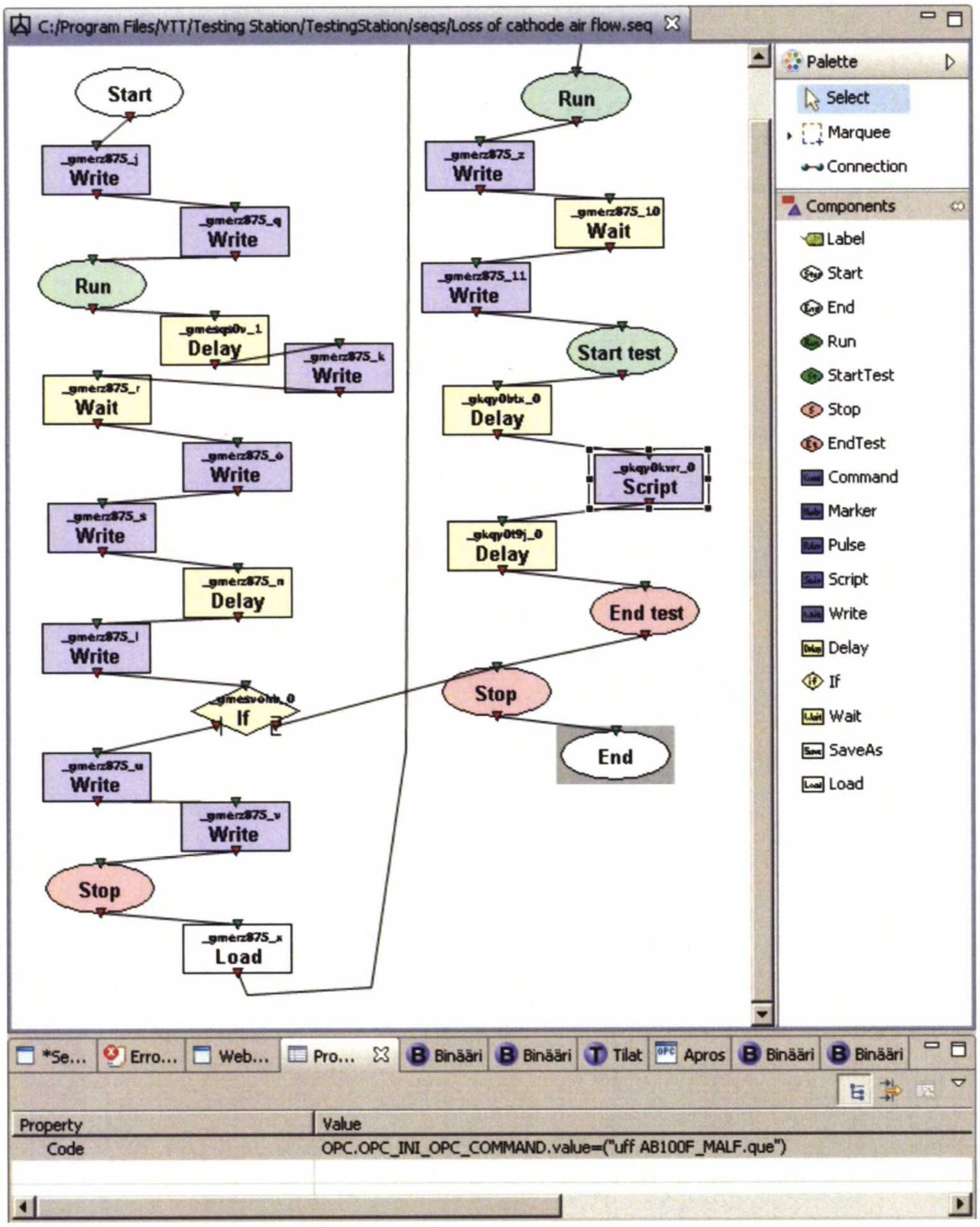

Figure 4.12: A sequence in Instructor's Station.

The second "test" part of the sequence is constructed to create the situation described in the verbal test description and to allow PLC to function as it's supposed to. This is usually achieved by using a script block to execute a que-script, created using the aforementioned excel-sheet, that contains the necessary modi-commands, and a delay block that keeps the test running long enough. The use of script blocks and que-scripts is strongly advised as it allows the same sequence format to be used in nearly all the test runs.All sequences must also contain the Run, Start, Stop and End blocks.

4.5 Testing

Testing is considered to include the phases between selection of the test case (or run) to be performed and the final reporting based on the obtained test results.

The flow of the tests is illustrated on a general level in figure 4.13. It shows how commands and information are transferred within the platform during tests.

4.5.1 Test Execution

Testing can be done automatically, in which case all the runs under one case are activated and executed automatically in a row, or one run at a time. Automatic testing is started by selecting "automatic testing" from the test cases context menu. No other measures are needed. A single run can be executed by activating it using its context menu and then clicking the play button on IS control panel. In both cases one can follow the process through IS sequence editor and the visualizations.

When a run is activated, IS loads its initial state and the sequence. When the run is begun, IS executes the sequence one step at a time. Between the "Start test" and "End test" blocks all subscribed data is stored internally.

4.5.2 Analysis of Test Results

For each individual tested automation functionality, the system produces a binary signal that indicates whether the system has behaved

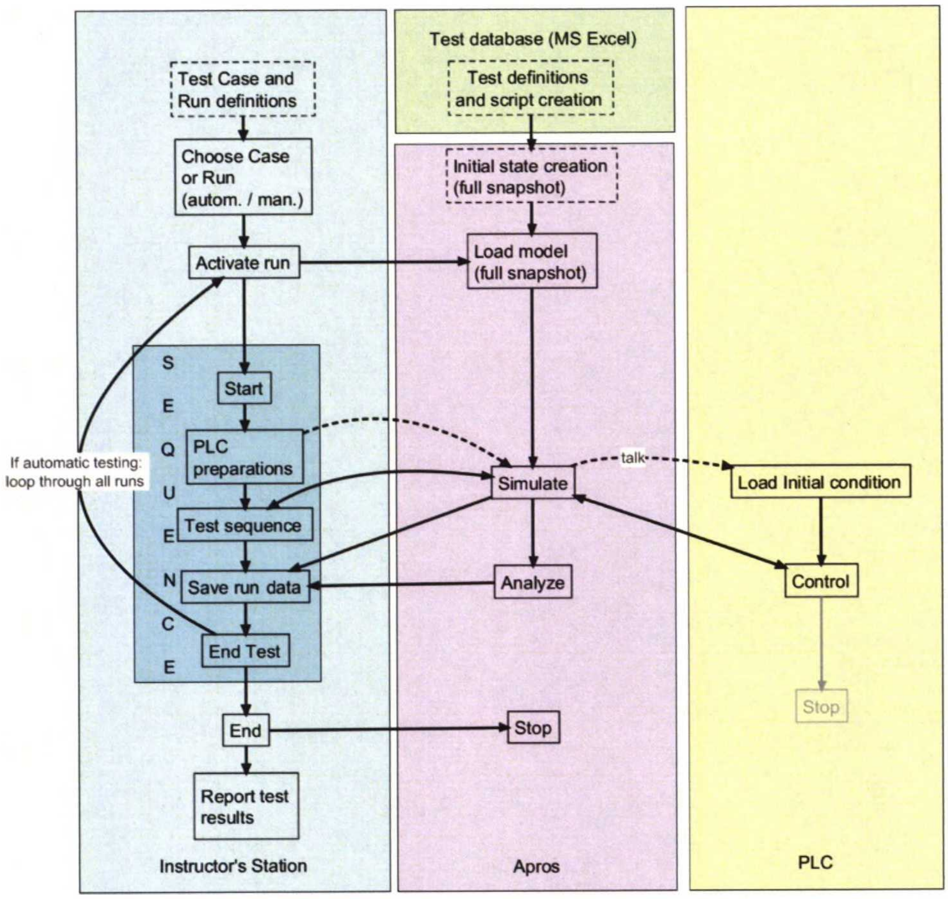

Figure 4.13: Flow chart of the actions performed by different parts of the platform during testing. The boxes with dashed borders represent actions performed during setup.

in an acceptable limits. Using these signals a general failure signal is calculated by using "OR" and flip-flop blocks. This general failure signal indicates whether something has gone wrong at any time during the test run.

These signal can be subscribed to and visualized in IS. From the visualizations it's easy to deduce if something went wrong and, if this is the case, what test(s) produced a positive failure signal.

4.5.3 Report Building

IS can automatically create a report from each run in html format. The report consists of all the visualizations of the run, the used sequence and some additional information. The functionality is still under development and does not provide all the planned features yet.

Main function of this reporting functionality at the moment is to provide an easy-to-read presentation of the most important test results. The person performing the tests can use it to quickly find out what individual test, if any, failed during the run.

To create the reports for all the runs, the user simply chooses "report" from the test runs context menu. IS saves the report and also individual plots of all the visualizations to the target folder, which by default is located in the root folder and named after the test run. This has to be done individually for all the runs at the moment. The resulting report consists of the graphs defined by user, including at least the binary signals for success or failure.

If this functionality is developed further and according to Wärtsilä's needs, it can become a tool that allows the automated building of high quality stand alone reports for various purposes, even for external distribution.

4.6 Alternative Automation Verification Solutions

This section briefly covers the different possible software options that could be used as parts of the platform. During this thesis no indication of a single clearly-better-than-others software was found. Given the current situation where Wärtsilä has already adopted Apros as its process simulation solution, software speculations can be limited to simulation control and analysis softwares and additionally to static verification softwares.

An extensive, but not complete, list of different products usable for software verification, especially static verification, can be found in Wikipedia's Software Testing Portal [3]. These programs use mathematics to deduce whether the code was written correctly. One widely recognized example is the Polyspace kit developed by the company MathWorks [5]. Though an important part of software verification, static methods cannot cover all the requirements for PLC unit verification. Dynamic methods must be applied.

For dynamic testing of the PLC unit a few products exist. Companies National Instrument (NI) and MathWorks have, for example, developed environments that allow the control of other products, such as simulators and control units through OPC servers. The environments are called LabVIEW and Matlab, respectively [4, 7]. Apparently, the simulator models could also be created using these programs but as fully functional Apros models already exist, there's little motivation to consider this.

A company called MYNAH has developed a complete simulation suite called MiMiC that includes the capabilities of Apros and Instructor's Station combined. In other words, it includes tools for simulation model creation and simulation control and can be used for functional testing of the automation system. According to the product website the system is built for software acceptance testing (SAT) and operator training systems (OTS) [6].

MathWorks programs, especially Matlab, are used widely and Matlab is the de-facto standard software for mathematical analysis. As an extremely flexible, multi-purpose software it can be considered the main alternative solution to replace the Instructor's Station software used in this thesis work. It's compared to IS in the next section.

4.7 Alternative Platform Based on Matlab® and Apros

This section describes an alternative platform solution. The basic testing idea and the protocol would remain the same.

Matlab is a multi-use numerical calculation software developed by MathWorks [4] and is capable of performing analysis and simulation tasks. It is a de facto standard tool throughout the world and has been used by the VTT staff to control Apros runs [1]. The main strength, weaknesses, opportunities and threats of using Matlab as a testing control application are presented in table 4.1.

Matlab and Apros are connected via an OPC server using Matlab's OPC Toolbox, which is commercially available. The realization in Matlab is composed of a tag group that contains references to Apros variables [19]. These references are, with a few exceptions, updated continuously during simulation and allow real time data transfer to and from Apros. The construction of this kind of connection closely resembles the one used in IS also. The main difference is in the way the connection data and parameters are stored: in Matlab all the information is contained in script-like text format in Matlab m-files whereas IS stores the same information in different types of internal objects.

Matlab, as an analysis software by character, allows a wide range of numerical analysis to be performed on the collected data. Should the testing protocol at some point require advanced mathematical analysis to be performed, it would be advisable to research Matlab's use more thoroughly. At this point however, as the required analysis is very simple by nature, Matlabs analysis possibilities offer little extra value for the platform. Using Matlab would also eliminate the need for analysis logic to be implemented in Apros but the benefits of this are, at this point, minimal.

Strengths	Weaknesses
• Non-visual sequencing & configuration • Integrated analysis applications and good reporting functionalities • Flexibility • De facto standard position • Publicity	• Non-visual sequencing & configuration • Requires skills and knowledge of Matlab • Price of licenses
Opportunities	**Threats**
• Powerful analysis functionalities in case advanced mathematical analysis is needed • Could replace X-connector • Use as a tuning instrument for PID controllers, for example	• IS may, through development, become a superior software for this particular purpose. • Dependency on OPC. • License price hike.

Table 4.1: SWOT analysis on the use of Matlab as a testing control application.

Chapter 5

Case Studies

To test the basic functions and usability of the developed protocol, three case studies were performed.

5.1 Testing of System Response to Emergency Shutdown Activation

5.1.1 Subject of the Test

The first case studies how the system's fuel feed reacts to the emergency shutdown (ES, for short) signal activated by the unit's operator pressing down an emergency shutdown button. The activation of the ES signal should cause the fuel feed to shut down within a time window of 15 seconds.

5.1.2 How the Test Was Performed

The Apros model of the system contains a binary signal module acting as the emergency shutdown button. This signal was activated by configuring the run sequence in IS to contain a script block assigned to run a script that activates the signal.

The Apros model also contains an analog signal for the hydrogen flow measurement result. Its value was measured by the test logic and compared to the limit value of 0.01 kg/s. The value was chosen to be above

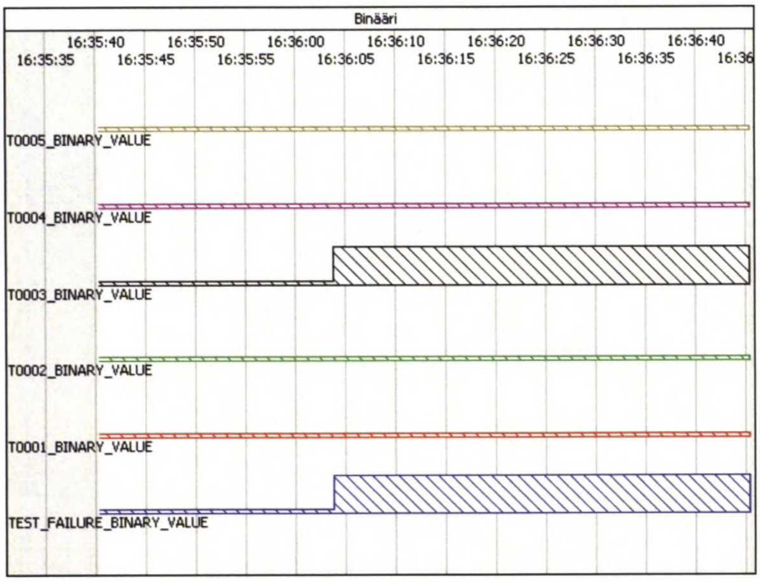

Figure 5.1: Test results for the system reaction to emergency shutdown activation. The results indicate a failure in the system's response but this was later realized to be the result of incorrect limit values.

zero to avoid false failures caused by calculation inaccuracies. The activation of the emergency signal was used as the trigger.

5.1.3 Results and Analysis

Figure 5.1 shows the results of the test run. It can be seen that the test has failed (in this case this test's ID number was T0003). This has also caused the general failure signal to activate. The rest of the visualized tests indicate expected behavior.

However, as the reason for the failed test was examined, it was found out that the limit values were set incorrectly and caused a false failure indication. After correcting the limit values the test was rerun and produced acceptable results.

5.2 Testing of System Response to Loss of Cathode Air Flow

5.2.1 Subject of the Test

This case studies how the system's hydrogen feed reacts to low cathode air flow. The amount of time the system has to react was not defined in the FAT listing so an ad-hoc value of 110 seconds was used for demonstration purposes.

5.2.2 How the Test Was Performed

The loss of cathode air flow was caused by triggering a malfunction of the air blower responsible for cathode air flow. This malfunction caused the blower to stop blowing. This was done in similar fashion as in the previous case.

The test logic was triggered by the cathode air flow process value becoming less than the minimum allowed value for it.

5.2.3 Results and Analysis

Figure 5.2 shows the results of the test run. It can be seen that the system functions correctly (in this case this test's ID number was T0001). The rest of the visualized tests also indicate expected behavior.

5.3 Testing of System Response to Its Transfer to Ventilation State

5.3.1 Subject of the Test

This case studies how the system reacts to its transfer to ventilation state. Transfer to this state protects the SOFC stack in case certain operation conditions are not met. The activation of the ventilation state should cause the fuel feed to be shut down within 15 seconds of the state transfer. In addition, the hydrogen feed should be at least 0.32

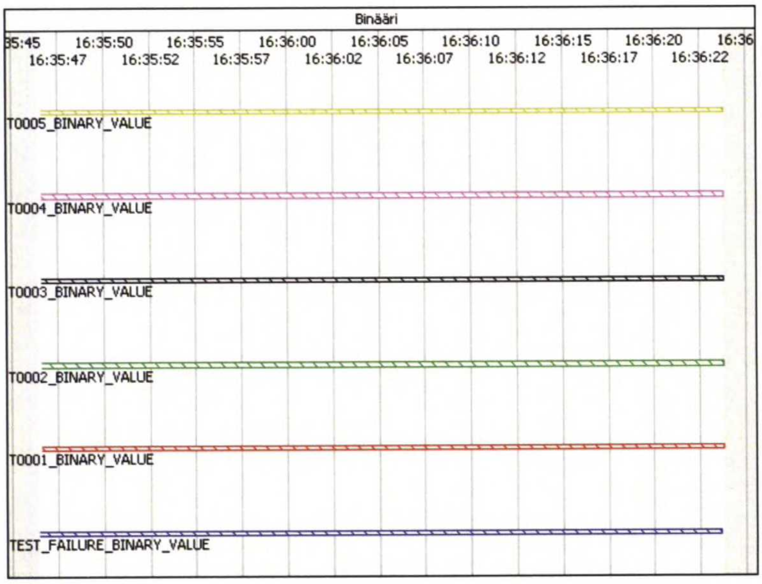

Figure 5.2: Test results for the system reaction to Loss of cathode air flow. The results indicate that the system performs correctly.

kg/s and the nitrogen feed at least 1 kg/s, both within 90 seconds from the transfer.

5.3.2 How the Test Was Performed

The system was "fooled" to transfer to ventilation state by causing reformer temperature measurements to give out simulated measurement values signaling a too low temperature in the reforming unit. This was done in similar fashion as in the previous cases. The activation of the ventilation state, indicated by a "system state" analog signal produced by the PLC unit and transferred to Apros, was used as the triggering signal.

5.3.3 Results and Analysis

Figure 5.3 shows the results of the test run. It can be seen that the system functions correctly (in this case these tests' ID numbers were T0002, T0004 and T0005). The rest of the visualized tests also indicate expected behavior.

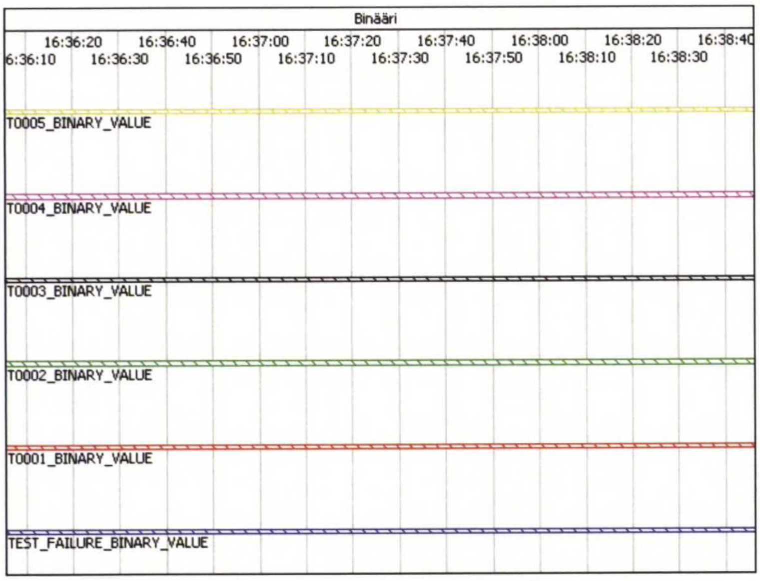

Figure 5.3: Test results for the system reaction to Loss of cathode air flow. The results indicate that the system performs correctly.

Chapter 6

Discussion and Conclusions

6.1 Discussion

The tools developed in this thesis support the tester's work and help in debugging the tested automation system. They cannot, however, completely compensate for rigor and carefulness of the tester and system developers.

There are several sources for possible errors. Some of them could possibly be eliminated through different measures including automated error recognition contraptions or subprotocols. The amount of work to create these kind of measures in large extent might, on the other hand, be very large and therefore it might be easier and / or more efficient to hunt the errors manually. One useful tool in this task is the Apros command prompt which prints a warning message in case the commands given to Apros are critically flawed. It does not, however, recognize all faulty commands.

The possibility of human error cannot be eliminated completely. The biggest risk factor lies in the incorrect definition of test parameters in the test database as the large number of tests increases the probability of a mistake.

6.2 Suggestions for Further Development

6.2.1 Suggestions for Further Development of the Testing Protocol

As the protocol is basically a set of instructions to be followed during a testing process, it must be kept up to date related to the system it was built for. As the protocol and platform develop and they spread into wider use, there's a risk that they will become too rigorous causing loss of organizational efficiency. Another risk is that the protocol evolves into a too complicated form and therefore loses it's functionality and becomes more of a nuisance than the useful tool it's meant to be. These risks can be avoided by giving enough control over the protocol and it's local use to the different teams or other organizational entities using it.

All in all, the protocol should be kept up to date and modified according to needs and experiences. It is advisable to test the protocol and how it serves the purpose it was created for now, as it has not been spread to wider use and is therefore easier to update. This will also make it more easily accepted among it's future users as they don't have to deal with "children's diseases".

6.2.2 Suggestions for Further Development of the Testing Platform

The main question regarding the developed testing platform is what program should be used to control the test runs. Apros is very likely to be the most suitable simulation software in the future also - mainly due to it's established position in Wärtsilä's fuel cell development. Rebuilding the models using a different simulation software would be a strenuous task and the training needed to accustom the staff with the new software would consume resources. Without clear foreseeable advantages, there's little reason to change Apros to a different software.

The realization of connections between the programs is dictated by the properties of the used programs and there's very little to be improved there, if anything. The biggest problem related to connections is the necessity of routing information exchange between IS and the PLC unit through Apros. At the moment there seems to be no solution that would allow a more direct configuration.

Instructor's Station on the other hand is a program still under development. It has very promising features and is already suitable for use as a run controlling and report building software in this developed testing platform. However, it still suffers from bugs and underdeveloped functionalities. If the bugs can be fixed and the functionalities developed or tailored to better suit Wärtsilä's needs, it could become clearly the best software for Wärtsilä's test automation purposes. The main benefit here is that it's being developed for, and by people very familiar with, Apros. Therefore there's great potential for a seamless connection between the two programs. The pairing of the two products is also the reason behind the fact that, should Wärtsilä decide to use another simulation software, there's a risk IS will become redundant.

The following list describes the most critical / beneficial development targets for IS from the viewpoint of fuel cell automation testing and Wärtsilä.

- Analysis tools for test analysis. At the moment IS offers no (working) tools for data analysis. At the simplest the possibility to write a test result into a variable inside IS and to present this result would be a great improvement. This shouldn't be too much of a problem as IS already offers the possibility to subscribe to different Apros variables. Therefore the analysis functionality could be achieved just by offering a tool for comparing two subscribed variables.

- The development of reporting tools for test result reporting. IS already offers a tool for reporting runs. At the moment, though, the tool is in practice capable of only printing all the visualizations created for the run into a single html-file. The possibility to include the aforementioned to-be-developed analysis results and other pieces of information would be very beneficial. From the viewpoint of IS's flexibility as a test control software the possibility of user defined report layouts would be a great improvement. This could be achieved with an interface that would allow placement of different test data modules onto a template.

- A direct link for controlling the PLC unit. This is not solely dependent on IS.

- Bug fixes and program reliability.

During this thesis work only one test template type was created in Apros. Even though it is very versatile and can be used in testing most imaginable situations, it's almost certain that other types of tests must be performed. This will in turn require the creation of new template types. If, or even when, this situation occurs, it's important to design the new templates to be as versatile as possible and to document how they work. The corresponding additions must also be made to the database for documentation and modification script creation purposes.

As the developed platform hasn't yet spread to a wider use within the company, it's advisable that Wärtsilä regularly considers whether the platform has a strong future within the company. Alternatives, such as the Matlab-based platform presented, exist and they should be considered. If, for example, it seems that Fortum or VTT are giving up the IS project or are taking it to a direction not serving Wärtsilä's needs, the Matlab based solution becomes more beneficial. Long term scenarios if IS's future should be examined.

6.3 Conclusions

The main results of this thesis work are the protocol and the platform created for the purpose of testing Wärtsilä's SOFC units' automation control units. These have been verified in several test cases. In addition, a user manual for the protocol and platform was written and several suggestions for further development produced.

The protocol directs the testing procedures and ensures that, if followed, all necessary things are taken into account in testing the automation system. The platform provides the necessary set of tools for automated testing and therefore reduces the active work time required to perform the tests significantly. The developed protocol and platform are presented in detail in chapters 3 and 4, respectively. Based on a literature survey it is reasonable to say the work is up-to-date with other comparable tools.

The protocol is readily usable with different automation systems, should such be taken into use. The platform can also be modified to work with different systems by updating the connections between it's different parts (softwares) to suite the new automation units data transfer needs. It is probable that the test definitions would require massive revision but this would need to be done anyways and cannot be considered to result from the testing platform or protocol.

The platform is fairly easy to keep up to date, even though its commissioning will require work on behalf of Wärtsilä. Its use is also easy to learn with little practice, even for a person who is not readily familiar with Apros or Instructor's Station. The IS is able to control even complicated runs using an intuitive graphical interface. Apros test templates and models need to be dealt with only in the case a completely new type of a test or when new initial states for the runs need to be created. The recreation of all initial states must be done when the model is modified. This work can however be automated, using IS for example, which reduces the work related to it. Otherwise only the test database needs to be updated and the modification scripts created and run.

One of the main benefits of the developed testing method is, that the tests are by default always on and independent of the test runs. Therefore all the defined tests are automatically performed in every situation thus adding to the reliability of the tests.

Work is still needed to improve usability of the software used, and especially to define all the necessary tests, but the protocol and platform are ready to be used in testing already.

Bibliography

[1] Discussions with vtt's antti pohjoranta, January - May 2011.

[2] Discussions with wärtsilä's staff, January - May 2011.

[3] http://en.wikipedia.org/wiki/portal:software_testing, June 2011.

[4] http://www.mathworks.com/products/matlab/, July 2011.

[5] http://www.mathworks.com/products/polyspace/, June 2011.

[6] http://www.mynah.com/products/mimic, June 2011.

[7] http://www.ni.com/labview, June 2011.

[8] http://www.processvision.fi, July 2011.

[9] www.apros.fi, May 2011.

[10] www.fuelcellstandards.com, May 2011.

[11] E. Achenbach and E. Riensche. Methane/steam reforming kinetics for solid oxide fuel cells.

[12] A. Eisentraut. Sustainable production of second-generation biofuels. Technical report, International Energy Agency, 2009.

[13] E. Fontell. Wärtsilä fuel cell program presentation, May 2011.

[14] E. Fontell, T. Kivisaari, N. Christiansen, J.-B. Hansen, and J. Pålsson. Conceptual study of a 150kW planar SOFC system for CHP application. *Journal of Power Sources 131*, pages 49–56, 2004.

[15] L. Fulton, P. Cazzola, F. Cuenot, K. Kojima, T. Onoda, J. Staub, and M. Taylor. Transport, energy and co_2. Technical report, International Energy Agency, 2009.

[16] R. A. Huggins. *Energy Storage*. Springer Science + Business Media, LLC, 2010.

[17] J. Kiviaho, M. Halinen, M. Noponen, J. Saarinen, P. Simell, and R. Rosenberg. Solid Oxide Fuel Cell System Development in VTT. *Journal of Fuel Cell Science and Technology*, pages 392–396, 2007.

[18] P. Laakso, M. Paljakka, P. Kangas, A. Helminen, J. Peltoniemi, and T. Ollikainen. Methods of Simulation-assisted Automation Testing. 2005.

[19] MathWorks. Matlab r 2009 b user manual.

[20] J. Saarinen, M. Halinen, J. Ylijoki, M. Noponen, P. Simell, and J. Kiviaho. Dynamic Model of 5 kW SOFC CHP Test Station. *Journal of Fuel Cell Science and Technology*, pages 397–405, 2007.

[21] S. C. Singhal and K. Kendall. *High Temperature Solid Oxide Fuel Cells: Fundamentals, Design and Applications*. Elsevier Ltd, 2003.