ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

NANYANG TECHNOLOGICAL UNIVERSITY

SECS/GEM MESSAGE DISCOVERY FOR RAPID EQUIPMENT INTEGRATION

SECS/GEM MESSAGE DISCOVERY FOR RAPID EQUIPMENT INTEGRATION

LIANG ZHIQIAN

LIANG ZHIQIAN

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

2007

2007

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

SECS/GEM Message Discovery for Rapid Equipment Integration

Liang Zhiqian

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Master of Engineering

2007

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Acknowledgement

I would like to take this opportunity to show my greatest thanks to Professor Dr. Ling Keck Voon and my supervisor Dr. Anton Aendenroomer for their invaluable guidance, help, and encouragement.

I would also like to express my sincere thanks to the SIMTech (Singapore Institute of Manufacturing Technology) Automation and Embedded System Team for their generous help and assistance.

Lastly, I would give my special thanks to my wife Wu Lihuan for her warmest care and always being my biggest support.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table of Contents

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Abstract

This project aims to analyze and suggest a solution to alleviate the problem of slow and tedious equipment integration process in the semiconductor manufacturing industries. The outcome is an intelligent software system which would allow the semiconductor equipment manufacturers to deliver their newly developed equipment more quickly and the end-users to start integrating of the "unknown" equipment with the host stations at a much earlier stage. The proposed system is designed to assist in the discovery of the SECS/GEM interface of semiconductor equipment, and to generate an application program to establish the communication between the host station and the target equipment. The development of the Message Discovery System, which can discover the SECS/GEM equipment interface is discussed in detailed in the report. The system consists of two main components, the Message Discovery System, which could automatically discover the SECS/GEM equipment interface, and the Code Generator, which generates the right interface program for the host station to control the discovered equipment. This project will put its focus on the design and development of the Message Discovery System. In this report, a self-learning expert system used in the Message Discovery System will be described in detailed. The architecture designs and high level modeling of the Message Discovery System using the UML notation will be elaborated. Finally, results of a preliminary test will be shown to demonstrate that the completed modules are working, and the design concepts verified. A paper for this project, "Message Discovery in SEMI Communication for Rapid Equipment Integration", has been published in the 2006 IEEE International Conference on Industrial Informatics.

List of Figures and Tables

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Abbreviations

waferfab	wafer fabrication plant
semicon	semiconductor
SEMI	Semiconductor Equipment and Material International
SECS	SEMI Equipment Communication Standard
GEM	Generic Equipment Model
HSMS	High Speed SECS Message Services Single Session
SECS-I	SEMI Equipment Communications Standard Part 1
SECS-II	SEMI Equipment Communications Standard Part 2
OBEM	Object-Based Equipment Model
API	Application Program Interface
GUI	Graphical User Interface
TCP/IP	Transmission Control Protocol/Internet Protocol
DA	DiscoveryAlgorithm
RDF	Resource Description Framework
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language
UML	Unified Modeling Language
OWL	Web Ontology Language
URI	Uniform Resource Identifiers
URL	Uniform Resource Locator

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 1 Introduction

1.1 Background

In recent years, semiconductor industry has been growing rapidly, and technologies used in semiconductor manufacturing have become more and more advanced and complex. In the past few years, the industry has moved towards the 300 mm wafer fabrication, and may very soon move to 450 mm waferfab (wafer fabrication plants). As the technology and the process of semiconductor manufacturing become complicated, newer and more advanced models of equipment come into play, making the integration of equipment increasingly crucial.

In order to continually increase the productivity, the winning strategy for semiconductor manufacturing companies is to implement highly efficient shop-floor automation with the capability of moving into factory-wide computer integrated manufacturing systems in the future. In front-end semiconductor manufacturing, there are a limited number of very big players who would invest huge capital to build waferfabs which take years to complete. They carefully organize and plan their activities in order to integrate their equipment (e.g. lithography or etching equipment). In this case, equipment integration is normally not an issue: the vendors and contractors who collaborate in building a waferfab would consider integration as one of the key activities in the project. However, backend semiconductor equipment manufacturers do not normally have the luxury of this systematic process of integrating many different equipment in a waferfab. There are various reasons for this; one of them is that backend equipment will usually not be installed during the initial phase when the waferfab is built.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

There are a large number of small equipment manufacturers who have to compete on features and time-to-market in order to survive in the competition. They would align their resources towards these goals rather than allocating resources to equipment integration. The eventual customers and their integrated setup are often unknown at the time of equipment design and assembly. As a result, when these equipment are manufactured and delivered to the waferfab, it will usually require the equipment users tedious effort to setup and integrate the new equipment into their factory networks. This project therefore aims to suggest an automated way of equipment integration to achieve the goals of rapid deployment, so that the equipment manufacturers can benefit from a faster time-to-market and the end-user from a shorter and painless ramp-up time.

Equipment System Integration

To provide a common foundation for the equipment interfaces, the semiconductor industry has developed various standards for production performance measurements, and communication between equipment and host, such as the SEMI standards: SECS/GEM (described in the later part) for communication. However, the equipment based on these standards is not usually compatible and has various drawbacks [2]. For example, many factories are facing the difficulties of data extraction from different equipment. Only certain data can be retrieved automatically, and some data must be typed in manually by operator or even cannot be retrieved.

However, from a central cell controller managing a group of equipment, one wish to monitor the status of the equipment, retrieve yield information, update on alarm conditions, download production data, etc. Therefore, system integration plays an important role in semiconductor manufacturing because equipment at all levels are integrated into a uniform network so that manufacturing information could be managed in the central host controller, which leads to benefits to the waferfab in automatic material handling and complete fabrication management.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

SEMI communication standards

In order to standardize the equipment in different aspects within the industry, a set of equipment standards are created by SEMI (Semiconductor Equipment & Materials International), which is an international organization governing the SEMI standards. Part of the standards deal with communication and define a set of standard messages plus a set of user-defined messages. The SEMI standards related to communication between equipment and host are, E4, E5, E30 and E37. These standards can be categorized into three groups based on a layer structure:

- Equipment behavior: GEM (E30)
- Message format: SECS-II (E5)
- Message transportation: SECS-I (E4), HSMS (E37)

As this project is developed mainly based on SEMI Equipment Communication Standards (SECS/GEM), especially on SECS-II, a brief description of the SECS/GEM standards is given in the following pages, for a better understanding of this project.

SECS-I

The SEMI Equipment Communications Standard Part 1 (SECS-I) standard defines the message exchange, through a RS232 point-to-point communication, between the semiconductor equipment and the host station [3].

SECS-II

The SECS-II standard defines the content of messages to be exchanged between intelligent equipment and host. These messages are organized into categories of activities known as streams, where the streams contain specific messages called functions. A Stream is a category of messages intended to support related activities. A Function is defined as a specific message for a specific activity within a Stream [4]. All primary messages are assigned an odd-numbered Function code and the reply message Function is determined by adding one to the primary function code. Function code zero is reserved for aborting transactions. Some of the Stream and Function code

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

combinations are reserved while others are available for user definition. Table 1.1 below shows the Streams and Function allocations.

Stream	Function	Description
		Stream Not Used In Standard
0	0-255	Function not defined for stream 0
1-63	0-63	Reserved for this standard
64-127	0	
1-63	64-255	Available for user definition
64-127	1-255	

Table 1.1 Stream & Function Allocation

HSMS

As a replacement for the point-to-point connection as defined in SEC-I, the High Speed SECS Message Service (HSMS) defines the equipment communication on top of TCP/IP link.

GEM

The Generic Equipment Model (GEM) gives specifications for equipment integration and rapid configuration through standardization of message, dialogs, scenarios and equipment behavior states and state transitions into Harel state-charts [6]. GEM defined which SECS-II messages should be used, in which situation and what the resulting activity should be.

Figure 1.1 shows how the above SEMI standards are applied and interface with each other in a host-equipment model.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

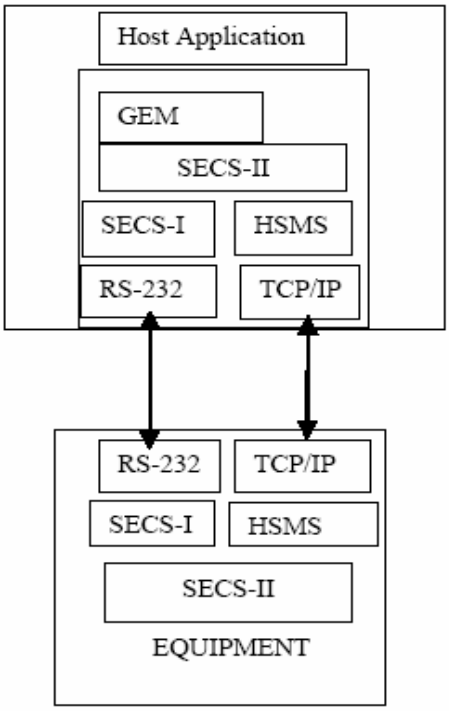


Figure 1.1 SEMI communication standards in a Host-Equipment Model

As can be seen in Table 1.1, a large portion of the Stream and Functions are allocated for user-defined messages. The user-defined messages are defined by equipment manufacturers and typically vary from vendor to vendor. The scope of this research is therefore about how to find out the extent of compliance of the equipment with the SEMI standard (i.e. how many of the standard messages is implemented) and to discover the user-defined messages implemented. The project aim is to develop an intelligent algorithm to discover the subset of standard messages and user-defined messages used in the equipment.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

1.2 Current Technology and Problem Faced

With the help of the SEMI communication standards, data transferring and equipment controlling between different equipment or between equipment and host controller have been standardized, which makes the equipment system integration easier than before. Nevertheless, in order to integrate a piece of new equipment into the factory network, it still requires a lot of manpower resources to study the equipment, and tedious effort to configure both the software and hardware on the factory host controller. In semiconductor industry, it's common that the equipment manual is sent to the equipment buyer quite some time after the equipment has shipped to the buyer. In order to integrate the new equipment into the factory network, the equipment integrators need to wait for the manual, then read it and understand the equipment configurations. On the other hand, they need to communicate with the factory administrators, so that they can customize the host controller to interface with this equipment correctly. As a result, the current problem faced by the industry is the considerable amount of time and effort spent before the equipment can start running after its arrival to the user.

Therefore, a software system for semiconductor equipment plug-and-play integration, which this project is proposing, is most desired and having a great value, as a rapid equipment deployment and shorter ramp-up time mean great values to the industry.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

1.3 Objectives of the Project

The main objective of this project is to analyze and suggest a solution to solve the problem of slow and tedious equipment integration process in the semiconductor manufacturing industries. The suggestion leads to an intelligent software system which would allow the semiconductor equipment manufacturers to deliver their newly developed equipment more quickly and the end-users to start integrating the new equipment with the host controllers at a much earlier stage. This software system aims to ease the difficulties of equipment integration by using software intelligence to automatically discover the SECS-II messages supported by the equipment and generate the right interface program for the host controller to interface with the equipment. This project gives the general design of the proposed solution but only focuses on the development of the Message Discovery System.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

1.4 Contributions

During the project period, the objectives of the project were successfully achieved. A solution for achieving easy semiconductor equipment integration is proposed. Detailed design and development of the Message Discovery System have been carried out.

Below are the major contributions of this project and work completed on developing the Message Discovery System:

- Successfully developed an expert system, which harness the knowledge gained from the SEMI communication standards to solve the problem faced in equipment integration. Established a framework to encode this knowledge into the expert system, which can be used to discover the SECS-II messages defined in "unknown" equipment.

- Successfully applied Goal-Driven Reasoning Methodology to the SECS-II messages discovery application. Instead of exhaustively search a large possibilities of SECS-II messages, Goal-Driven Reasoning allows the search to be carried out in a more efficient manner and saves computing resources.

- The HSMS-SS standard is implemented in the HSMS Stack, and a SECS-II message parser is implemented in the SECS-II API as C++ libraries.

- Design, implementation of the Equipment DataBase as a MySQL data base filled with initial equipment data has been completed.

- Design, implementation of the Rule Base in an ASCII data file filled with initial rules has been completed.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

1.5 Organization of the Thesis

In Chapter 1 of this report, a brief introduction of the background and project objectives, as well as the major contributions of our work are given. In Chapter 2, design concepts of the proposed solution, and the methodology adopted in the Message Discovery are elaborated. In Chapter 3, system design and implementation details of the Message Discovery System, as well as results of a system performance test will be shown. Finally, project conclusions and recommended future works will be presented in Chapter 4.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 2 The Proposed Solution

2.1 Design Overview

As mentioned in the previous chapter, although the SEMI SECS/GEM standards are designed for better equipment integration, they are still far from a fully plug-and-play integration. Currently in the market, there are a large number of equipment suppliers, but their software solutions are still proprietary, due to the reason that they can have many equipment-specific user-defined SECS-II messages in the equipment. Each supplier can have his own combination of SECS-II messages to achieve a specific function of his equipment.

Moreover, every manufacturer (who needs to integrate equipment from different suppliers for his production) customizes his cell controllers to manage a group of equipment in a unique way. This introduces many difficulties coming from the addition of new equipment (and their user-defined messages) under this particular cell controller due to the different classes of equipment (and different suppliers).

In order to integrate the new equipment into the factory network, equipment vendors most of the time need to revise the equipment's host-interface software (with the cell-controller) accordingly. Therefore, it would be of tremendous benefit for the system integrator if he has the ability to automatically detect the equipment communication interface and if he were provided with tools for automatic code generation to generate the interface code at the host-side.

As a result, this project suggests a software solution, which consists of the SECS-II Message Discovery System and the Code Generator, to ease the above difficulties in equipment integration. Figure 2.1 shows the functional design of the proposed

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

solution.

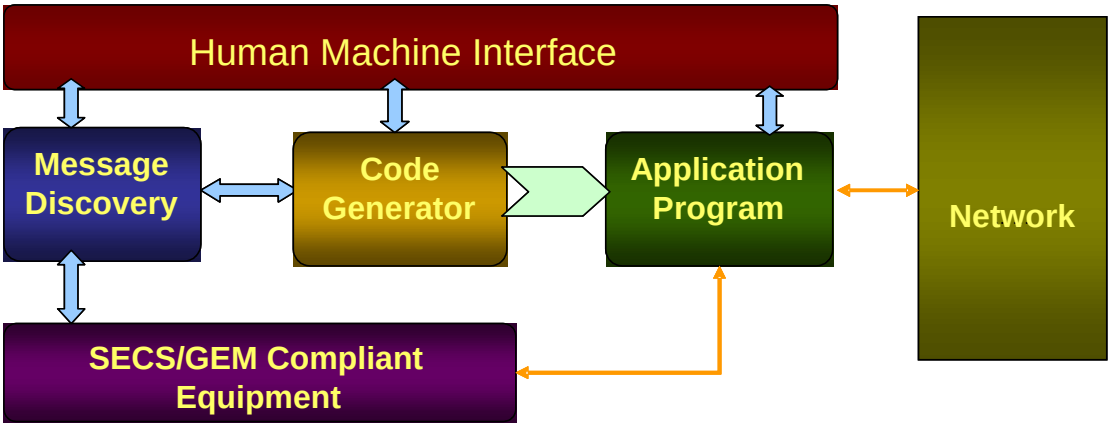


Figure.2.1 Functional Design of the Proposed Solution

The above diagram shows the basic information flow of the whole system. When the system sees a piece of new or "unknown" equipment, the Message Discovery module will be activated to probe the equipment and discover the equipment characteristics. Then by utilizing the knowledge discovered, the Code Generator generates an Application program, through which this piece of new equipment can be easily integrated into the factory network and controlled by the factory host station.

Overall, the solution proposed consists of two main functional blocks: "Code Generator" and "Message Discovery". The scope of this project focused on the design, implementation and evaluation of the Message Discovery System.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

2.2 Intelligence in the Solution

2.2.1 Possible Approaches

As we known, the reason why we need to spend time and effort on equipment integration is because at the time when the equipment is delivered, the host station doesn't know what are the SECS-II messages defined in this equipment, which means it doesn't know what are the functions and capabilities of this equipment. Hence in order to have an automatic equipment integration solution, an intelligent way to discover all the SECS-II messages in the equipment is required. To achieve this, applying artificial intelligence (AI) in message discovery would be the most likely approach. In another words, AI could be deployed in a software system to automatically discover the set of SECS-II messages that are defined in the equipment.

In the area of Artificial Intelligence, the following approaches have been mostly adopted to solve different types of practical problems.

- **Expert System** to provide expertise on a problem in specific knowledge domain. An example will be a medical diagnostic program. By inputting the patient's symptoms, the program can infer the correct diagnosis from the expert knowledge in the database, so that can help the doctor for a faster diagnosis on the patient. [10]
- **Neural Network** for data prediction. As an example in the stock market, people use neural network to determine the average premium (discount) the market is currently allocating to particular industries, and then use that standard in an industry-by-industry neural network analysis designed to determine which stocks are trading below their market value.[15]

- **Data Mining** for information retrieval from a large volume of data. A simple example of data mining, often called Market Basket Analysis, is the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

information analysis for retail sales. If a clothing store records the purchases of customers, a data mining system could identify those customers who favor silk shirts over cotton ones. [16]

In order to select an appropriate AI technique, we need to first understand the characteristics of our target problem. The problem we need to solve is how to discover the set of SECS-II messages defined in a piece of new equipment during the host-equipment communication. So, we have the following four main characteristics about the target problem:

- The target problem is within a specific knowledge domain, which is the domain of semiconductor equipment integration. We are actually targeting a problem that is usually solved by the equipment integration experts.

- The information flow during the communication is discrete and event-based and consists of data packets with content depending on the situation related to the event. It does not resemble closed-loop systems with a number of input and output channels, which can be modeled either by differential equation or neural network. Hence we cannot apply mathematical formulas or computational methods on the data.

- Information collection for different types of equipment might not be practical during the project period. Since the information is proprietary to a particular equipment vendor, so we don't have the luxury of rich information about various types of equipment from different vendors.

- The process of the discovery can be interactive. We can have additional information input through background information by user (e.g. we know already we are dealing with a Wire Bonder and even some specific

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

information about this Wire Bonder etc.)

One of the requirements of adopting either Neural Network or Data Mining techniques is that a large amount of data is needed to train the Neural Network or to retrieve useful information before such an intelligent system could function correctly and provide useful information. On the other hand, knowledge can be learned and built up during the operation of an expert system. Therefore, the above characteristics suggest that an expert system approach is the most appropriate for developing the Message Discovery System.

2.2.2 Expert System

Human experts are able to solve high level problems because they have deep knowledge (both facts and rules) and strong practical experience in a particular domain. In general, an expert system is a computer system that facilitates solving problems in a given field or application by drawing inference from a knowledge base developed from human expertise [10].

Below is a typical architecture of an expert system:

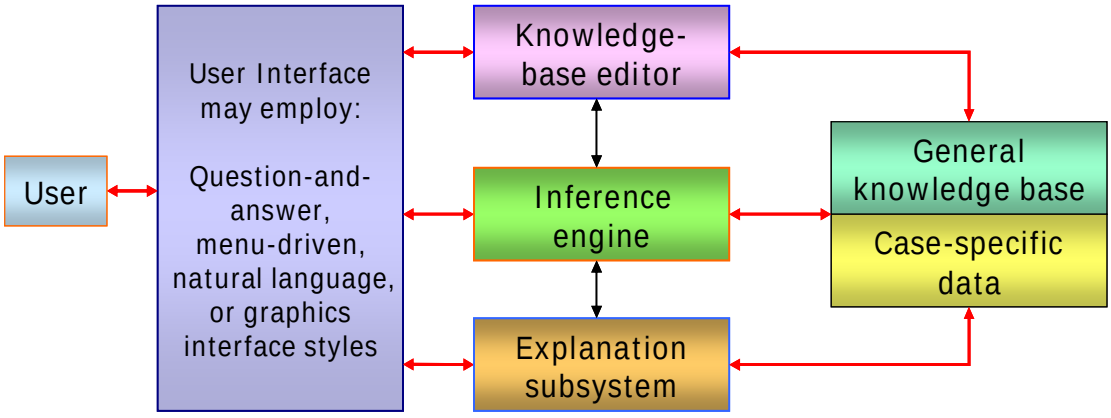


Figure 2.2 Architecture of an expert system

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Rule-based expert system

Rule-based expert systems represent problem-solving knowledge as if……then…… rules. This approach lends itself to the architecture of Figure 2.2, and is one of the traditional techniques for representing domain knowledge in an expert system. It is also one of the most natural, and remains widely used in practical and experimental expert systems.

In a rule-based system, the heart of the system consists of three core modules: the *knowledge base*, the *case-specific data* and the *inference engine*. The *knowledge base* contains the specific domain knowledge represented as *if……then……* rules. There are two portions in the rules: the premises of the rules, the *if* portion, corresponding to the condition, and the conclusion, the *then* portion, corresponding to the action. When the condition is satisfied, the expert system takes the action of asserting the conclusion as true.

The expert system must also keep track of the *case-specific data*: the facts, conclusions, and other information relevant to the case under consideration. This information is separate from the general *knowledge base*.

The *inference engine* applies the knowledge to the solution of actual problems. Simply speaking, the *inference engine* applies rules from the *knowledge base* onto the information from the *case-specific data*, and performs the resulting conclusions.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The above architecture of a rule-based expert system will then be our design guideline of the Message Discovery System, discussed in the following sections of this chapter.

2.3 The Message Discovery

Based on the logic of a rule-based expert system, the Message Discovery System is designed to solve the problem like a human expert in this field. The general strategy is to keep on asking questions, in order to discover the unknown information by analyzing the responses, and at the same time to define more questions. As in our case, the common language among semiconductor equipment is the SECS-II messages. Hence asking questions is actually implemented by sending SECS-II messages to the target equipment. Building up knowledge is achieved by analyzing the SECS-II reply messages and defining more SECS-II messages to send. Finally, the unknown information to discover is the particular SECS-II messages that are supported by the target equipment, which is the aim of this project.

2.3.1 Confining the Target Domain

For meaningful knowledge acquisition the target domain must be clearly identified within boundaries (confined). In our approach we confine the domain as: semicon equipment which complies with the SECS-II, HSMS and GEM standards. The reasons below explain why.

In order to apply the expert system strategy in the discovery, a bidirectional communication must be established between the target equipment and the software system. In another word, the target equipment must firstly understand SECS-II messages and be able to response in SECS-II messages to the messages that sent to them. Hence our first Prerequisite is the target equipment must be SECS-II standard compliant.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Under the SECS-II standard, Message Stream 9 (refer to Table 1.1) provides a method of informing the message sender that the message has been received cannot be handled. When this happens, the stream 9 messages indicate either a message fault or a communication fault has occurred. With the implementation of this Stream 9 messages, semiconductor equipment will be able to properly response to any SECS-II messages sent to them. On the contrary, equipment without the Stream 9 messages implementation will not be able to response to the SECS-II messages sent to them, if those messages are not specifically defined in the equipment. As a result, SECS-II Stream 9 messages implementation is another prerequisite for the target equipment of our proposed system.

Since the last decade, SECS/GEM standards have been widely adopted by most of the equipment manufacturers; and under the GEM standard, every equipment is required to implement Stream 9 messages. What's more, equipment networks in semiconductor factories are usually built on TCP/IP, so HSMS compliance is a must for equipment need to be integrated into a factory network. Therefore, it's not too restrictive to confine our target equipment to those that are compliant with the SECS-II, HSMS, and GEM standards.

2.3.2 Knowledge Acquisition

Knowledge Acquisition is one of the key processes in the development of an expert system. An expert system is a program which solves the target problems like a human expert. So it is necessary for the expert system to gain enough knowledge of the problem domain, before the system operates. However, an advantage of expert systems is that they are built by progressive approximations, with the program's mistakes leading to corrections or additions to the knowledge base. In a sense, the knowledge base is "grown" rather than constructed [10].

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

In this project, the necessary knowledge that needs to be gained during the development consists of information, facts, and rules which are basically all in the SEMI standards. This knowledge is exact and strictly practiced by all the equipment suppliers: a supplier applies a subset of this knowledge into his equipment (the subset that is required to fully specify his equipment for later integration). Hence, once our expert system has possessed this knowledge, it is then able to communicate with and analyze any equipment in our target domain. On the other hand, the knowledge of configurations about different types of equipment and equipment from various suppliers is variable, but it can be "grown" while our system is maturing and capturing more information from different vendors, models etc. Therefore, as a pilot project, our knowledge base will be developed with extractable information from the SEMI standards and information about limiting types of semiconductor equipment.

In our approach the knowledge base consists of two "databases": the rule base and the equipment database. The rule base is built up by interpreting the rules originating from the SEMI standards and the equipment database is built-up by gathering information regarding different types of equipment (in terms of type, model, vendor etc) during the lifetime of our system.

2.3.2.1 Populating the Rule Base

In our design, knowledge extracted from the SEMI standards is stored in the Rule Base. The Rule Base stores Equipment-relationships, SECS-II message-relationships and constraint rules defined by the various standards and equipment suppliers. This information is useful for the system to identify the unknown SECS-II messages and the message content structures in the target equipment. The Rule-Base also can be updated by the user or by the program when there are new rules found or there are changes on the existing rules.

A Rule Base is the basic element of a reasoning system. The rules allow the system to

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

deduce new results from an initial set of data. For a simple example, one of the rules in our Rule Base is:

```
[(?e eq:HasStdSECSMsg eq:S1F15) -> (?e eq:HasStdSECSMsg eq:S1F17)]
```

This above rule is constructed in the format read by the Jena reasoning engine, which is adopted by the proposed system (the Jena module will be introduced later in Chapter 3). This rule means any equipment that supports SECS-II message S1F15, must also support S1F17.

There are many rules stored in the Rule Base, and they will be searched once the system has gained information from the equipment. If this information satisfies the condition of a particular rule, this rule will be fired, and extra information can be deduced. Implementation details of the Rule Base will be given in Chapter 3.

2.3.2.2 Constructing the Equipment DataBase

The Equipment DataBase is designed to store the case-specific data in an expert system, which in this project is the useful information from various equipment. In the Equipment DataBase, equipment is differentiated by categories, functionalities, vendors and models. Under each entry, besides the equipment category, models etc. there are also entries that capture the SECS-II messages that are supported and their message content formats.

The main purpose of the Equipment DataBase is to keep the historical data of the expert system, just like the database of a medical doctor accumulates by the experience gained, each time he sees a patient. Hence after each discovery process, the Equipment DataBase will accumulate the updated information of the newly discovered equipment.

The Equipment DataBase is an important support of our expert system. It provides the program with known information about equipment that has been discovered. By having the historical information, not only can the program straight away give out the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

characteristics of the target equipment (if there is an entry for it already), but also the program can apply some known information on the target equipment. For example, for equipment from the same manufacturer, the same equipment-defined SECS-II message will most likely use the same data formats in the message content.

Below is a diagram showing how the Equipment DataBase is conceptually designed:

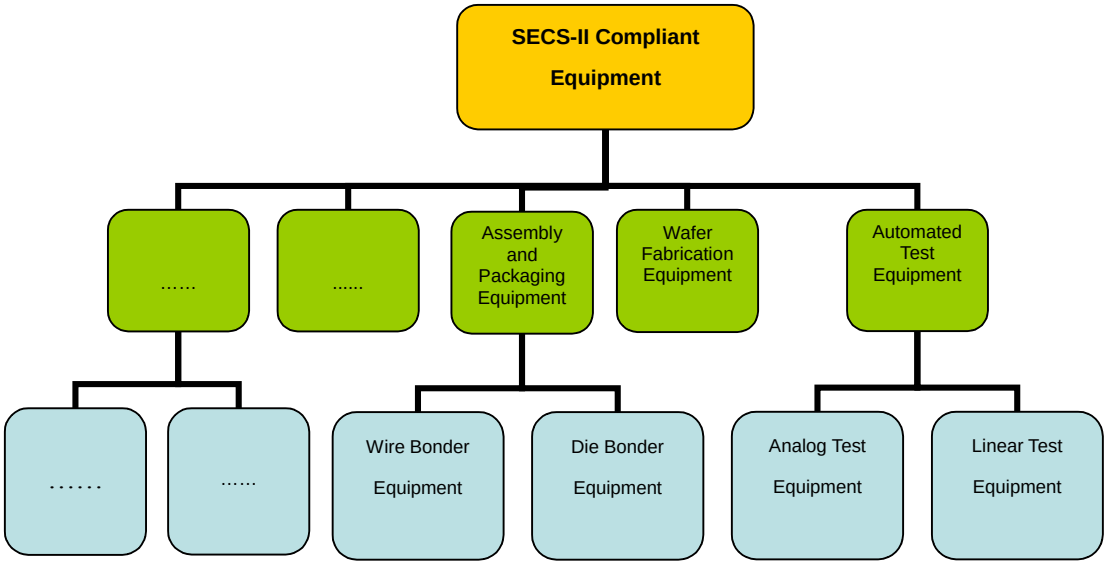


Figure 2.3 Equipment Database concept design

The Equipment DataBase is basically designed into an equipment category tree. In the database, equipment are classified into different categories, and different categories of equipment have different choices of SECS-II messages as well as different data format. The equipment tree can have as many branches as possible, and grow as the system discovers more equipment. Besides providing the necessary storage mechanism, the equipment database will play a role in the discovery process by providing additional information for the end-user in how the SECS-II messages are related to the actual equipment we are dealing with and the final code to be generated.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

2.3.3 Goal-Driven Reasoning Methodology

As a key process in developing an expert system, designing the reasoning method is equal to helping the system to find a way to the target. The reasoning method defines how the system will apply rules from the Rule Base on the information gathered.

In our design, the main reasoning technique used during the message discovery system is the Goal-driven Reasoning Methodology. In a goal-driven reasoning system, the goal expression is initially placed in working memory. The system matches rule conclusions with the goal, selecting one rule and placing its conditions in the working memory. This corresponds to a decomposition of the problem's goal into simpler sub-goals. The process continues in the next iteration, with these conditions becoming the new goals to match against rule conclusions. The system thus works back from the initial goal until all the sub-goals in the working memory are known to be true, indicating that the hypothesis has been verified.

Based on this method, the expert system firstly probes the target equipment with some initial messages (replies to these initial messages can tell which category the equipment is in, say for example bonding equipment: wirebonder or diebonder). Then by analyzing the information gathered from the equipment replies, the expert system makes an assumption on the equipment categories (coming back to the example: the assumption that the equipment is a wirebonder). In order to prove this assumption, the system will generate more messages to probe the equipment (same example: it will probe with messages that are common for wirebonders; if the reply is negative then the hypothesis is rejected and the equipment is classified as diebonder) and make a more accurate assumption based on the replies (a more accurate assumption would be a particular model under the diebonder category). In this way, the assumption on the equipment categories will be more and more exact, until all the SECS-II messages supported by the equipment have been discovered. By following this method, considerable amount of the discovery time will be saved as compared to sending the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

target equipment all the messages one by one.

More details of how the proposed system discovers the SECS-II messages in the target equipment are shown in the Discovery Process.

2.3.4 Discovery Process

Since our aim is to solve the problem of tedious effort spent by the equipment integrators trying to study the equipment configurations, this software system is therefore designed to automatically discover the functions and capabilities supported by the target equipment. For example, Equipment A supports the equipment standby function (S1F15 – OFFLINE, S1F17 – ON LINE), but Equipment B does not. When the two pieces of equipment are integrated into the factory network, the host controller must know that it can turn off Equipment A by sending message S1F15, but Equipment B on the other hand need to be switched off manually. Therefore, more specifically, the proposed system aims to discover all the SECS-II messages supported by the target equipment, so that the host controller could know exactly how to control the target equipment.

2.3.4.1 Preliminary Knowledge

In an expert system, problems are solved by applying the existing domain knowledge into the process of gathering the facts and data from the problem domain. Therefore, in order to apply the expert system design in the solution, a way of gathering information from the target equipment is necessary. In our design, the Stream 9 errors reply messages (refer to SECS-II standard) are used as the main channel of getting information from the target equipment. According to the SECS-II standard, when a SECS-II message reaches the equipment, if the equipment is not able to handle this

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

message, one of the Stream 9 messages will be replied to the message sender. The followings are the 2 Stream 9 messages that will be handled by the proposed system:

- S9F3 – Unrecognized Stream Type

  When this message is received by the system, it means the equipment does not recognize the stream type of the message that sent to it. For example, a piece of equipment doesn't support any message from stream 10. If the system tries to send it any message from stream 10 (e.g. S10F5 or S10F11 ……), the system will get a S9F3 message reply.

- S9F5 – Unrecognized Function Type

  When this message is received by the system, it means the equipment does not recognize the function type of the message that sent to it. For example, a piece of equipment supports message S10F5, but not S10F11. If the system tries to send it message S10F11, the system will get a S9F5 message reply. Here, an important implication of the reply is that although the equipment doesn't support S10F11, it supports at least one message from this stream; otherwise the reply would be a S9F3.

There are a few more Stream 9 reply messages defined by the standard, and they are about the data errors and timeouts of the messages. However, if the system has received a Stream 9 message other than S9F3 or S9F5, it means the message stream and function of the outgoing message are both supported by the target equipment.

Therefore, 3 types of replies can be expected when the system probes the target equipment with a SECS-II message. The equipment reply tells the system the outgoing message is supported, the stream type of outgoing message is not supported, or the function type of the outgoing message is not supported (refer to Figure 2.4). This is how and what information is gathered from the target equipment during the discovery process.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

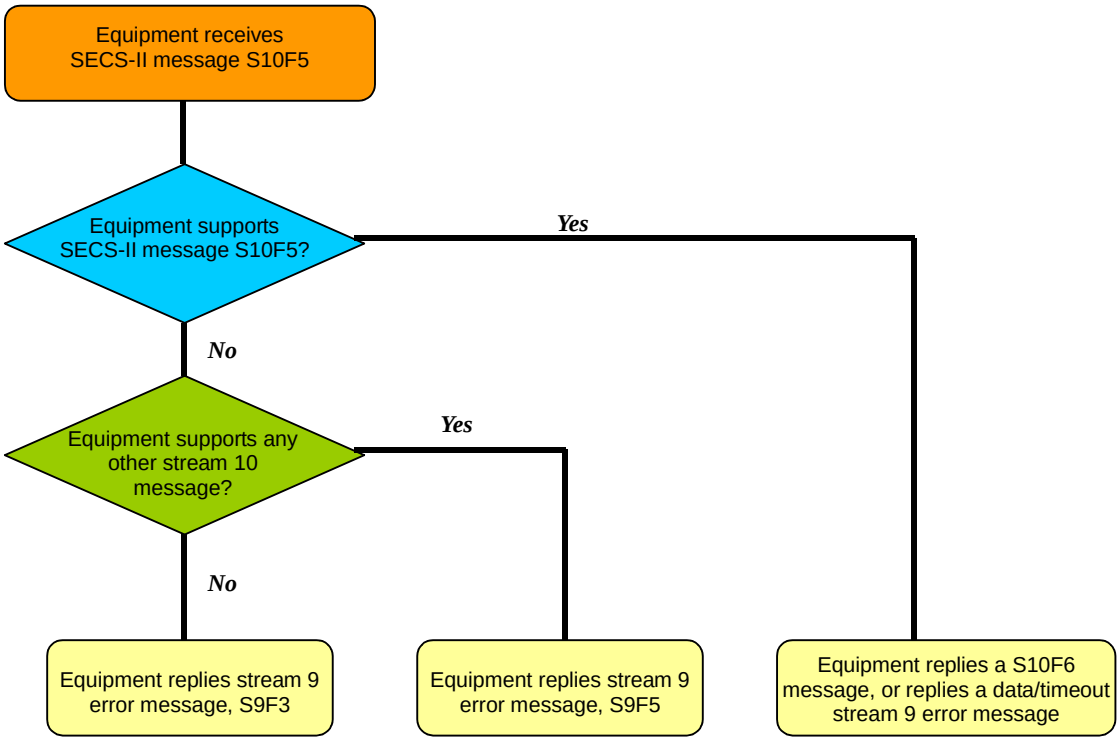


Figure 2.4 Possible Equipment Replies

2.3.4.2 Pre-discovery Setup

Setting up the Discovery System

First of all, in order to enable the Discovery System to run against new equipment joining the factory network, the necessary software and the discovery program should be installed on the cell controller or a computer connected to the same network.

When a piece of new equipment needs to be integrated to the factory network, it will be assigned a local network address (for a TCP/IP network, it will be a local IP address), a communication port if necessary, and as well as an equipment ID. This information is important for the program to setup a connection between the Discovery System and the target equipment.

Equipment Information Collection

To increase the discovery speed, the program user is encouraged to enter as much

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

information as he knows to the user interface. A snap shot of the program's configuration pages are shown below:

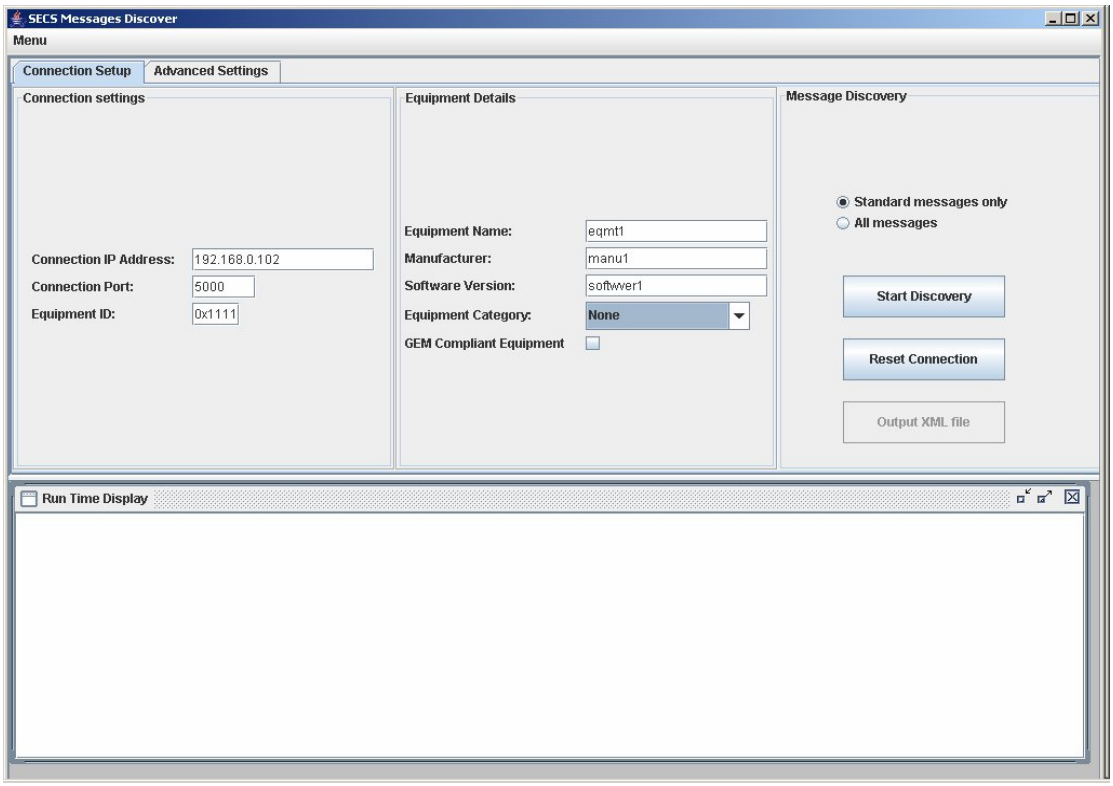


Figure 2.5 Program configuration for the discovery

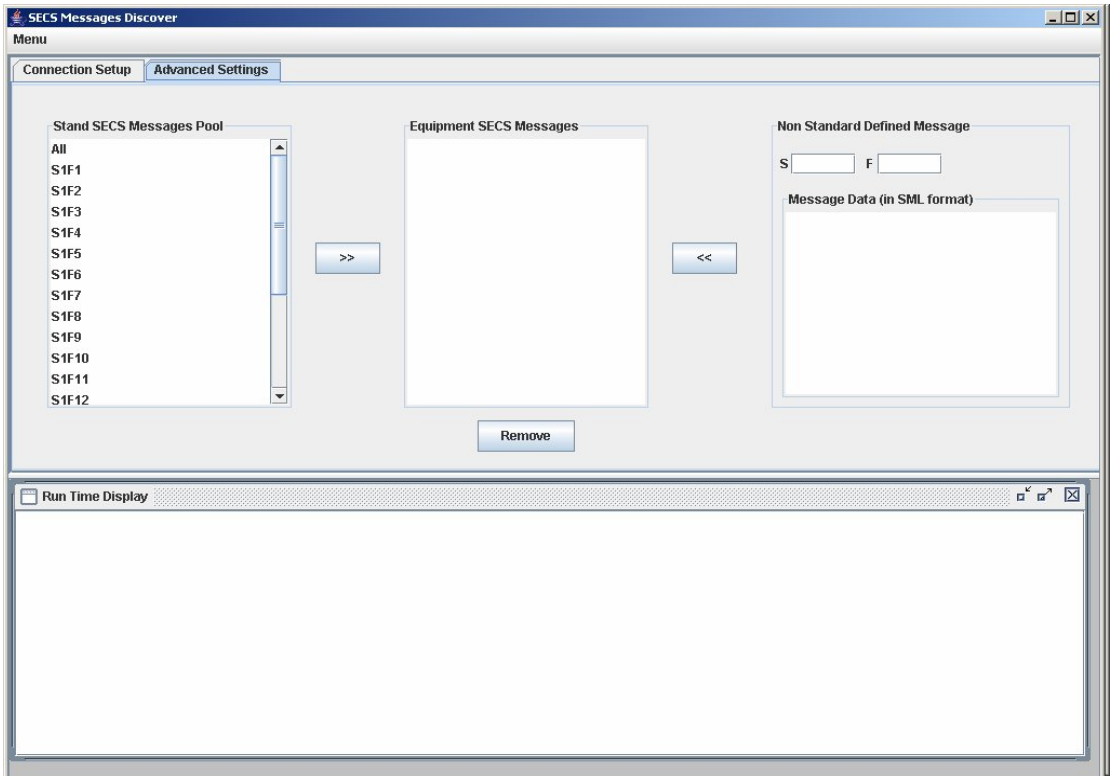


Figure 2.6 Advanced program configuration for the discovery

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Information, such as equipment category, SECS-II messages defined as well as the message content formats, will be used in helping the program to make a more exact assumption during the discovery, so that the discovery process can be shortened.

2.3.4.3 Discovery Initialization

Equipment Schema Generation

Before the discovery process starts, if the program user has keyed in some information about the target equipment, this information will be compiled into an equipment information holder called the Equipment Schema, which is actually an equipment object class in RDF (Resource Description Framework, a brief introduction about RDF will be given in Appendix D) representation. If no information is keyed in by the user, the Equipment Schema will be empty. Obviously, if the Equipment Schema is empty, no information processing needs to be done and the discovery will be started with default procedures.

The Equipment Schema contains the information that the user has keyed in. This information will be saved into different properties of this equipment object. For example, if the user input states that the target equipment is a Wire Bonder and has SECS-II message S1F3, S1F11, and S1F13, then in the Equipment Schema generated, the property "EquipmentCategory" has a value of "Wire Bonder", and the property "HasStdSECSMsg" has values of S1F3, S1F11, and S1F13.

In order to give a more practical understanding, a simple sample of the Equipment Schema in RDF format is provided in Appendix A for reference.

2.3.4.4 Sending Initial set of SECS-II Messages

For an empty Equipment Schema

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

If the discovery process is started without any user input, a set of predefined SECS-II messages will be sent to the target equipment. These predefined SECS-II messages contain one message from every SECS-II message streams. If the target equipment is not even compliant with SECS-II standard, there won't be a proper reply from the target equipment. However, if the target equipment is SECS-II compliant, and has defined some SECS-II messages in several streams, the discovery program will get a standard error message reply for every SECS-II message sent whose stream is not defined in the target equipment. For example, if the predefined set are 8 SECS-II messages from stream 1 to stream 8, and the target equipment has defined messages from stream 1 to stream 3, so for every message from stream 4 to stream8 sent, the discovery program will get a standard error message reply to indicate that the target equipment does not define SECS-II message from that stream.

As a result, by sending this predefined set of messages, the program is able to firstly verify whether the target equipment is SECS-II compliant, and secondly discover the SECS-II message streams defined in the target equipment.

This set of predefined SECS-II messages consist of all first message in every stream, which is:
S1F1, S2F1, S3F1, S4F1, S5F1, S6F1, S7F1, S8F1, (Stream 9 is used for error messages, not included in the predefined set), S10F1……S127F1.

For a non-empty Equipment Schema

If before the discovery process, the user has keyed in information about the target equipment, for example, if the user keyed in that the target equipment is a wirebonder, besides the predefined set of messages, those SECS-II messages that are common in wirebonders will be also sent to the target equipment. Hence after this procedure, the Equipment Schema will contain whether the common messages for wirebonders are defined in the target equipment, and the Equipment Schema will also state the message streams that are defined.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

More importantly, the more information the user can input before the discovery, the more accurate and exact the initial assumption can be. In another word, more time will be saved on the discovery process.

2.3.4.5 Applying Inference Based on the Equipment Schema

Making Assumption

After the previous steps, the Equipment Schema will at least have some information about the target equipment, so rules in the Rule Base can now be applied by the Inference Engine. The Equipment Schema will be passed to the Inference Engine and all rules in the Rule Base will be applied on the information in the Equipment Schema. Then an assumption will be made on the equipment categories in the Equipment Database.

For example, the following steps show how the Inference Engine made an assumption that the target equipment is a Wire Bonder machine under the equipment category of Assembly and Packing Equipment.

- When a piece of Wire Bonder equipment was connected to the system, an Equipment Schema was generated by the program and the predefined set of messages was sent to the equipment. After analyzing the reply from the target, the equipment has been verified to have SECS-II message stream 1, 2, 5, 7, 10, and this information was then stored into the Equipment Schema.

- By applying the rule,

  [(?e eq:HasSECSMsgStream eq:Stream1), (?e eq:HasSECSMsgStream eq:Stream2), (?e eq:HasSECSMsgStream eq:Stream5), (?e eq:HasSECSMsgStream eq:Stream7), (?e eq:HasSECSMsgStream eq:Stream10) -> (?e eq:BasicCategory eq:AssPackingEqmt)]

  the inference engine can verify that the target equipment has defined message

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

stream1, stream2, stream5, stream7, and stream10, and thus can be categorized to the Assembly and Packing Equipment. This finding will now be added into the Equipment Schema.

- When applying another rule

  ```
  [(?e eq:BasicCategory eq:AssPackingEqmt), (?e eq:NOSECSMsgStream eq:Stream12)
  -> (?e eq:AssumeEqmtType eq:Wire Bonder)]
  ```

  the inference engine can verify that this Assembly and Packing Equipment doesn't have stream12. Now an assumption is made that the target equipment belongs to a Wire Bonder type. This assumption will be verified by sending more messages to the equipment that are common to all wirebonders. Following the same reasoning the next assumption to be made will try to classify this wirebonder into types of wirebonder such as ultrasonic or thermosonic wirebonder,. After verification of this assumption, the next assumption could be related to equipment type and model.

In another word, to make an assumption based on the Equipment Schema is actually classifying the target to the possible equipment categories on the equipment category tree (refer to Figure 2.3).

In order to make an assumption a scenario list is required that configures the different scenarios of messages to be sent in all the steps mentioned above.

Defining a Scenario List

In order to verify an assumption made, more information needs to be collected from the target equipment. This information is basically in the form of SECS-II messages. Since different category of equipment has different combination of SECS-II messages, so to prove the assumption on a certain equipment category, we need to prove that the target equipment has defined a common set of SECS-II messages in this category.

At in the previous example, after an assumption of a Wire Bonder had been made on

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

the target equipment, more information was needed to collect from the equipment. This was done by defining and sending out a Scenario List. A Scenario List is actually a list of SECS-II messages that common in the assumed equipment category. This list of SECS-II messages will be passed to the Test Engine, and action will be taken by the Test Engine to interpret and send out the actual format of these messages.

Let's refer to the previous example, after the assumption was made, rules were applied on the Equipment Schema again so that the program could gather more information from the target equipment. One of the rules was the following:

```
[(?e eq:AssumeEqmtType eq:Wire Bonder) -> (?e eq:TestSECSMsg eq:S1F3), (?e
eq:TestSECSMsg eq:S1F3), (?e eq:TestSECSMsg eq:S1F11), (?e eq:TestSECSMsg
eq:S1F13), (?e eq:TestSECSMsg eq:S1F15), (?e eq:TestSECSMsg eq:S1F17), (?e
eq:TestSECSMsg eq:S2F13), (?e eq:TestSECSMsg eq:S2F15), (?e eq:TestSECSMsg
eq:S2F17), (?e eq:TestSECSMsg eq:S2F23), (?e eq:TestSECSMsg eq:S2F29), (?e
eq:TestSECSMsg eq:S2F31), (?e eq:TestSECSMsg eq:S2F33), (?e eq:TestSECSMsg
eq:S2F35), (?e eq:TestSECSMsg eq:S2F37), (?e eq:TestSECSMsg eq:S2F39), (?e
eq:TestSECSMsg eq:S2F41), (?e eq:TestSECSMsg eq:S2F43), (?e eq:TestSECSMsg
eq:S2F45), (?e eq:TestSECSMsg eq:S2F47), (?e eq:TestSECSMsg eq:S5F3), (?e
eq:TestSECSMsg eq:S5F5), (?e eq:TestSECSMsg eq:S6F15), (?e eq:TestSECSMsg
eq:S6F19), (?e eq:TestSECSMsg eq:S6F23), (?e eq:TestSECSMsg eq:S7F1), (?e
eq:TestSECSMsg eq:S7F3), (?e eq:TestSECSMsg eq:S7F5), (?e eq:TestSECSMsg
eq:S7F17), (?e eq:TestSECSMsg eq:S7F19), (?e eq:TestSECSMsg eq:S10F3), (?e
eq:TestSECSMsg eq:S10F5)]
```

From the above rule, the program knew that in order to prove the assumption, this list of SECS-II messages was required to verify their existence. This list of messages then became a Scenario List and was passed to the Test Engine for sending them to the target equipment.

Applying Rules and Making Inference

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The Rule Base with reasoning rules is a key component of the Expert System. It helps the Expert System to infer more information based on the existing. It also helps the Expert System to make decisions and draw conclusions. In this project, the rules will be applied to the Equipment Schema through a Reasoner in the Inference Engine. Whenever the Equipment Schema is updated with some new information, it will be passed to the Reasoner. The Reasoner will firstly retrieve all information in the Equipment Schema and apply all rules in the Rule Base on this information. The Reasoner then fires all the rules whose conditions are satisfied and add these rules' conclusion information into the Equipment Schema. Finally, because the Equipment Schema is updated, the Reasoner will iterate the steps of applying all the rules until no more rules can be fired.

Hence in this way, inferences are made and the Equipment Schema is updated with more information to help the Inference Engine on defining new Scenario Lists or drawing conclusions.

For example, if the Equipment Schema shows that the target equipment supports message S2F1, a simple rule **`[(?e eq:HasStdSECSMsg eq:S2F1) -> (?e eq:HasStdSECSMsg eq:S2F3)]`** could fire, then the Equipment Schema is updated with S2F3, and the system will know the target equipment also supports S2F3.

2.3.4.6 Completing the Discovery

After the list of SECS-II messages on a Scenario List have been sent to the target equipment, the program will wait for the replies to identify whether these messages are defined in the target equipment. This new information will be added to the Equipment Schema, and a new assumption will be made on which equipment category that the target equipment most likely belongs to. Then a new Scenario List will be created to prove the new assumption. These discovery procedures continue on until all SECS-II

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

messages supported by the equipment has been discovered. At this time, the target equipment will be proved belong to an existing equipment category or a new one which doesn't exist in the Equipment DataBase.

After the discovery process, information of the target equipment will be stored into the Equipment DataBase, so that it can be used in the future discovery process. In addition to that, if the target equipment belongs to a new equipment category, the user maybe asked to create a new equipment category, and rules in the Rule Base will be updated accordingly. As a machine self learning feature, the Rule Base updates itself with rules extracted from the Equipment Schema for this new equipment category. By doing this, the Inference Engine will be able to make the correct assumption when it sees equipment from this category again in the future.

2.3.4.7 Generating a RDF Report for the Target Equipment

Information for the Host Controller

Finally, after the whole discovery process is done, information about the target equipment is discovered and stored by the system. This information will be used to customize the host controller so that the equipment can be correctly integrated with the host controller. However, in order to achieve an automatic equipment integration solution, the host controller must be able to be automatically modified by the Discovery System based on the equipment information. In modern software technologies, code generation is the most common way for customizing and modifying a software system automatically by itself.

In this project, the code generator is used to generate the customized communication interface between the host controller and the target equipment. By sending different SECS-II messages to the equipment, the host controller controls the equipment to achieve different functions. Hence in order to customize the communication interface

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

to a particular piece of equipment,

- Firstly, the code generator needs to know the set of SECS-II messages that are defined in this equipment.
- Secondly, properties of the equipment, such as equipment name, manufacturer, software version, equipment ID, network address, port number and GEM compliance, also need to be presented to the code generator. Especially for GEM compliant equipment, by knowing which standard functionalities that the equipment supports, the code generator can use the corresponding routines, which results in reducing the complexity of the generated code and a faster communication time.
- Moreover, for non-standard equipment defined SECS-II messages, the data format of the message content is also important information to the code generator, so that the host controller is able to communicate using this message without any data error.

In the view of data representation, for an effective way of presenting information, the above data can actually be presented in the structure of an object.

Code Generation

On the other side, in code generation, XSL (eXtensible Stylesheet Language) is often utilized to generate program code from the Meta Data and the Model Program (as shown in the figure below).

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

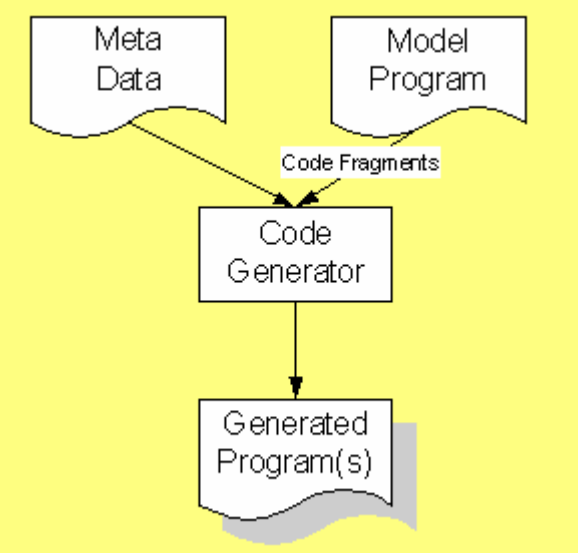


Figure 2.7 Structure of a code generation program

In this technology, XML (eXtensible Markup Language) is usually used to represent the Meta Data, which is used to describe the characteristics about the program to be generated. Because XML is excellent in representing structured data and having more portable, light weight formats.

RDF Report

Therefore, since the discovered equipment information needs to be represented in XML as well as in object structures, we found that the RDF (Resource Description Framework) representation is most suitable for our case. Because RDF provides an XML-based syntax for representing information, and more importantly, it's designed for describing resources in the format of objects. (A brief introduction about RDF will be given in Appendix D)

By generating a RDF report, the Discovery System passes the equipment information discovered to the code generator. This information includes equipment name, equipment manufacturer, equipment ID, network address, communication port, equipment type, GEM compliance, and most importantly the standard defined SECS-II messages as well as whether there's equipment defined SECS-II messages. Based on

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

this report, the code generator will be able to generate the interface program for the host controller to communication with the target equipment seamlessly. Until then, a successful equipment information discovery process is completed.

A simple sample of the RDF report is attached in Appendix B for reference.

Chapter 3 Project Implementation and Test Results

This chapter will describe the detail project implementation flow of the functional modules of the system, which includes the logical high level designs in UML as well as the information flow and interfacing techniques among all the modules. However, due to the complexities, detail coding process of all the modules are not covered in this chapter.

At the end of this chapter, a project test is shown with a real scenario from an actual piece of equipment, so that the design concept of the project and performance of the developed system can be tested. This test serves the purpose of evaluating the performance of the system and verifying the design concepts.

3.1 System Architecture Design

The Message Discovery System consists of several modules. These modules are separated into three functional groups, which are presented as three program layers.

- HSMS Layer, which has the TCP/IP Stack module and the HSMS-SS Stack module, handles the communication interfacing between the program and the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

target equipment. Since the system is designed to work on a TCP/IP communication link, a TCP/IP Stack is needed in the HSMS Layer in order to pack the data into TCP/IP packets. A HSMS-SS Stack is also needed, so that messages can be packed compliant to the HSMS protocol in order to allow the target equipment to correctly unpack the messages.

- SECS-II Layer, has the SECS-II API as the only module in it. Because all data sending to or receiving from SECS/GEM compliant equipment must be in the format of SECS-II messages, so a message interpreter is needed to translate the SECS-II messages into a language that the application program can understand. This SECS-II messages interpreter is called the SECS-II API module in this system.

- The last layer is the Application Layer, which contains the rest of the functional modules of the system: Test Engine, DiscoveryAlgorithm, SECS-II Messages, Equipment DataBase, and Rule Base. This layer is responsible for asking and getting replies from the target equipment (in the form of sending and receiving SECS-II messages); by applying rules and logic reasoning on the replies, it discover the equipment supported SECS-II messages; finally update the database and rule base if necessary (see Figure 3.1).

Figure. 3.1 shows the block diagram design containing all the modules and information flow of the Message Discovery System:

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

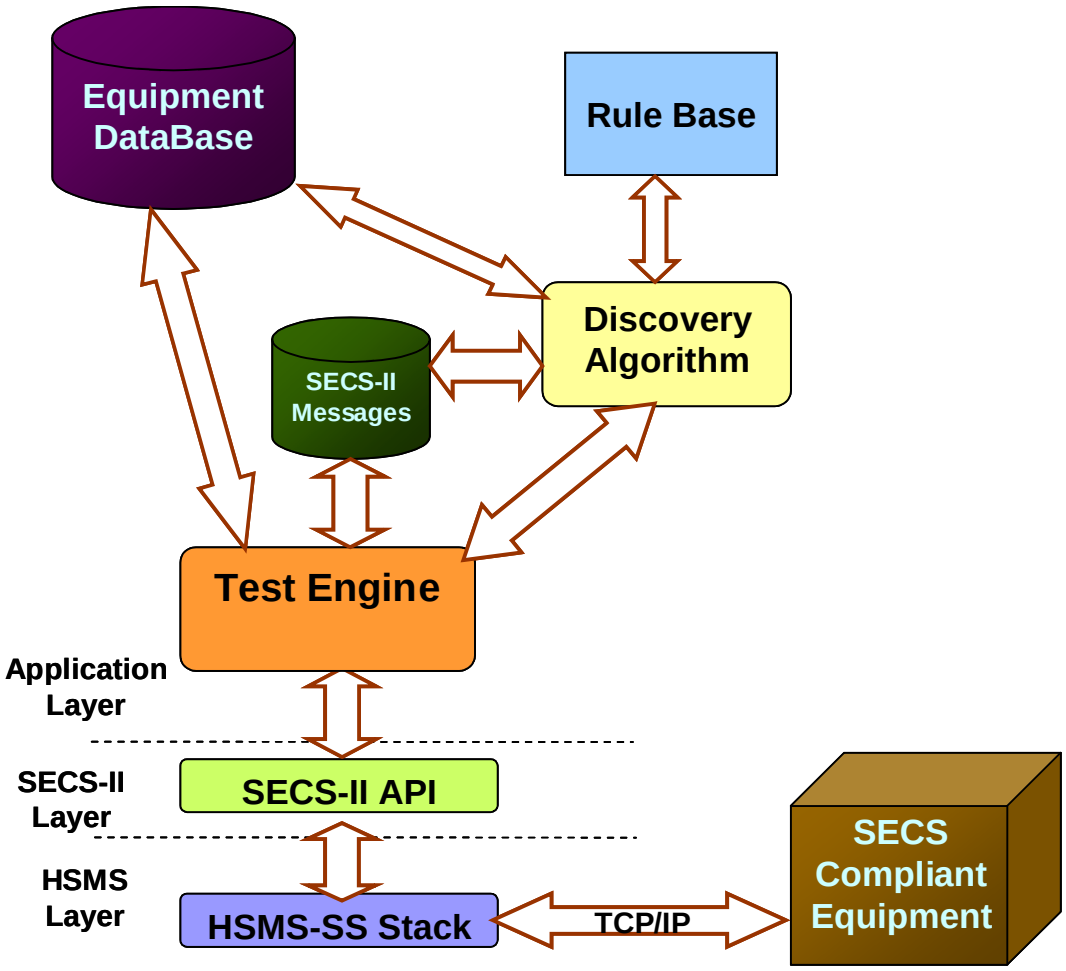


Figure 3.1 Block diagram design of the Message Discovery System

By referring to the diagram above, design objectives, functionalities of and interfacings between all the module blocks are explained in the following section of this chapter:

3.1.1 HSMS-SS Stack

High Speed SECS Message Services Single Session, or HSMS-SS for short, is a well-known standard widely adopted by semiconductor equipment manufacturers. HSMS-SS defines the data format of the SECS-II messages that are transferred between the equipment and the host over TCP/IP communication link.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

SECS-II messages that are sent out by the higher layer, will be packed with HSMS headers added by the HSMS-SS module, so that messages can be transferred through TCP/IP following the HSMS standard. Vise versa, SECS-II messages sent by the equipment, will be having an error checking, and their HSMS headers will be striped off before they are sent to the high layer of the system.

3.1.2 SECS-II API

As the only module in the SECS-II Layer, SECS-II API is an important program interface between modules in the Application Layer and those in the HSMS Layer. It works as a data format interpreter for the communication between the upper layer and the lower layer.

As mentioned previously, the target semicon equipment can only understand and send out plain binary data that are formatted under the SECS-II standard. In order to allow the Application Layer to have faster processing on these SECS-II messages, the SECS-II API module is designed to translate the SECS-II messages between binary data and C++ object code representations. SECS-II messages received from the equipment will be translated from binary codes to C++ objects here, so that modules in the upper layers can understand. While SECS-II messages sent by the TestEngine that are in the form of C++ object will be translated to binary codes, so that they can be understood and processed by the target equipment.

3.1.3 TestEngine

To increase the efficiency of the whole discovery process, it’s better to separate the message transmission process and message analyzing process, so that these two

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

processes can work concurrently. Therefore the TestEngine is designed, which mainly handles the delivery of messages between the DiscoveryAlgorithm (DA) and the equipment. Once it has received and understood a Scenario List (explained in section 2.3.4.5) from the DA, the TestEngine will form and send out the SECS-II messages specified in the Scenario List. Before the messages are sent out, the detail contents of the messages will be retrieved from the Equipment DataBase. Upon receiving the reply messages from the equipment, the TestEngine organizes them and stores them into the SECS-II Messages Database. In addition, a flag will be returned to the DA when the TestEngine has stored the messages, has finished a Scenario List, and when error occurs.

3.1.4 Rule Base and Equipment DataBase

Functionalities of the Rule Base and the Equipment DataBase, as well as how they are developed are explained in Chapter 2 (refer to section 2.3.2).

3.1.5 DiscoveryAlgorithm(DA)

Based on some known information stored in the Equipment DataBase and rules in the Rule Base, the DiscoveryAlgorithm discovers the defined SECS-II messages in the equipment by using a self-learning expert system. After the end of discovery process, the DiscoveryAlgorithm also updates the Equipment DataBase and the Rule Base, so that this equipment is remembered and the system doesn't need to run the discovery again when next time sees it.

In order to discover the unknown messages in the equipment, the DA will follow a strategy, and keep on sending, receiving messages to and from the equipment. After

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

some initial messages sent and replies have been received from the target equipment, the DA applies rules in the Rule Base on the replied information and draws some inference. It then makes an assumption based on the inference results, and defines a Scenario List, which contains a group of SECS-II messages, to verify the assumption. This Scenario List will be sent to the TestEngine, who will interpret the Scenario List and send the messages one by one to the equipment. After the reply messages from the target equipment have been received and stored into the Database by the TestEngine, the DA will start to analyze those received messages again. Meanwhile, the DA can define and send another Scenario List to the TestEngine, so that these two modules can work concurrently. By doing this, the system makes use of the time spent on waiting for the reply from the equipment.

3.1.5.1 Architecture of the DA

By taking reference from the architecture of a typical expert system, the DiscoveryAlgorithm is design into a similar structure. It contains an Inference Engine, a program interface to the Equipment DataBase, a program interface to the Rule Base and an interface to the user. A block diagram design of the DiscoveryAlgorithm is given below:

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

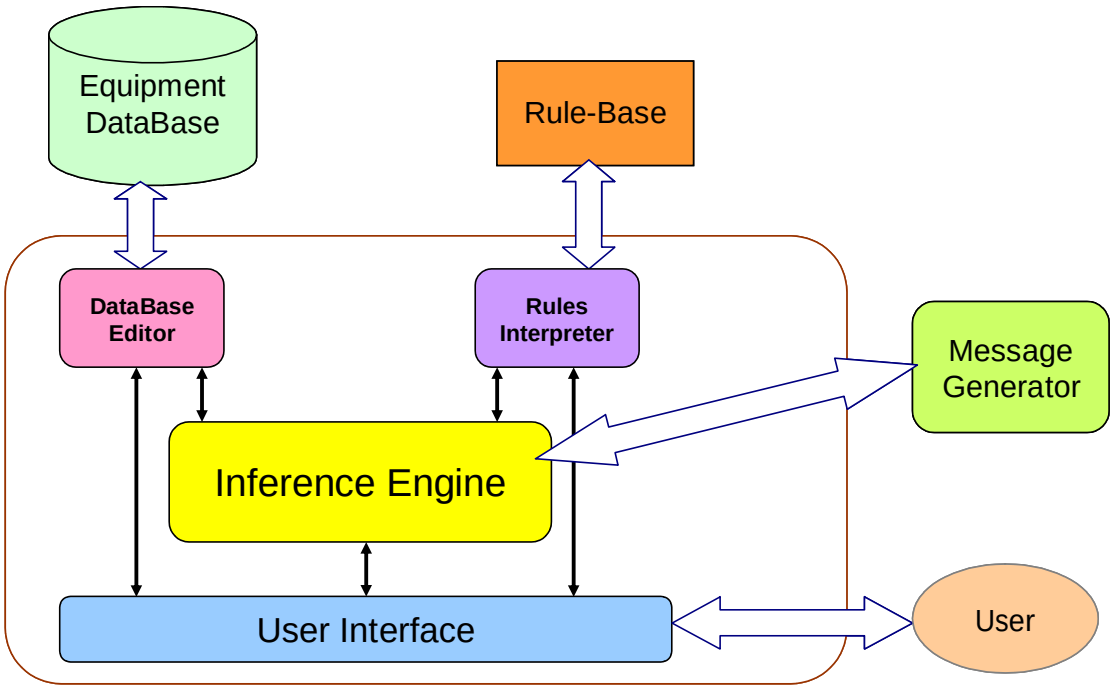


Figure 3.2 Block diagram design of the DiscoveryAlgorithm

3.1.5.2 Rules Interpreter

Being a program interface for the Rule Base, the Rules Interpreter module is designed to have the abilities of understanding the format of the rules, changing, adding, deleting the rules stored in the Rule Base. The Rules Interpreter will be called when there is a need for Inference Engine to apply the rules or edit the rules based on the data it received.

As the heart of a rule-based expert system, the rules in the Rule Base contain the knowledge of a particular application domain. The knowledge will be represented in the form of if……then…… rules, which are close to the human expression. This representation enables the system developers an easier way of updating the Rule Base in the future.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

3.1.5.3 DataBase Editor

The DataBase Editor is the program module that manages the adding, deleting, and editing of items in the Equipment DataBase. Different from the Rule Base, Equipment DataBase stores the equipment related information, such as equipment features and supported SECS-II messages of different kinds of equipment. The Inference Engine will keep track of the information in the Equipment DataBase so that it can refer to it or modify it when necessary.

3.1.5.4 User Interface

The User Interface is the necessary module for the system to interact with system operators. Through the User Interface, system operators are able to edit, add or delete rules in the Rule Base as well as entries in the Equipment DataBase. Most importantly, the User Interface allows the system to explain the reasoning result and show the discovery process to the operator.

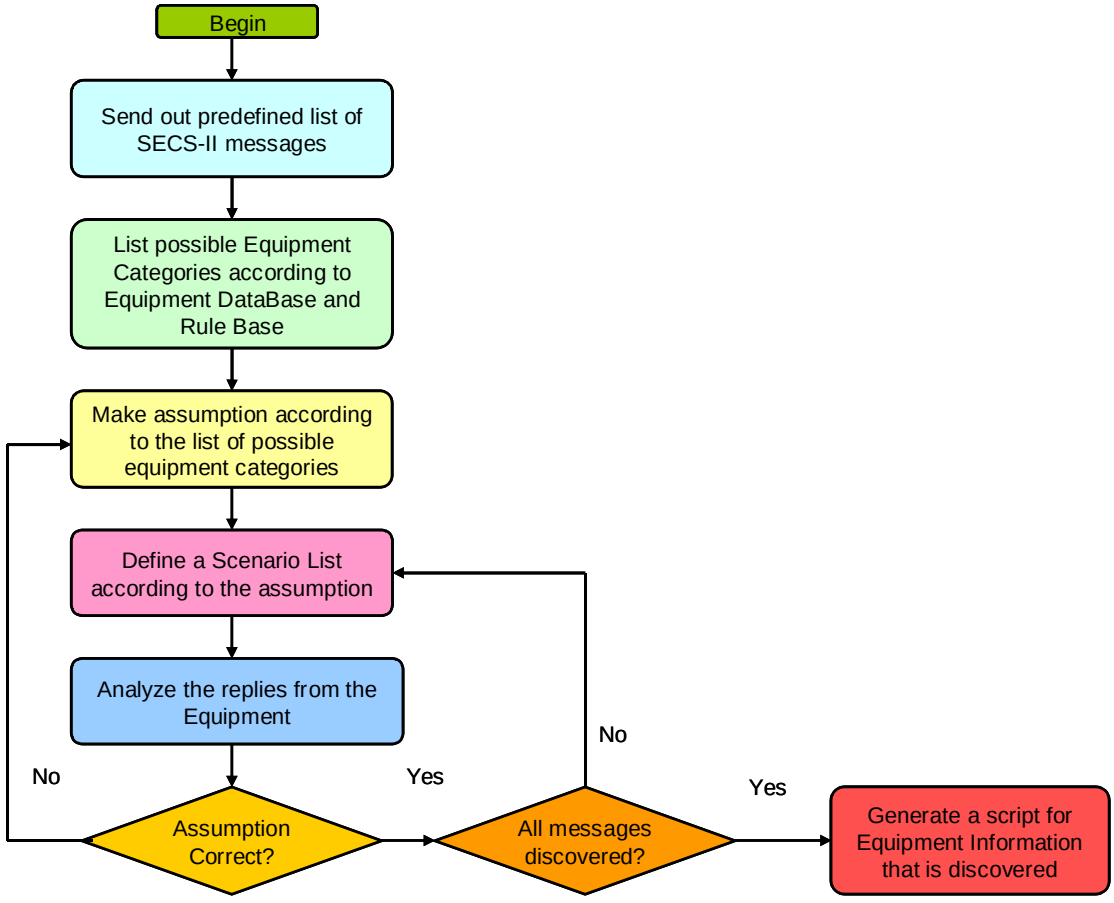
3.1.5.5 Inference Engine

The Inference Engine is the main data processor, and logical reasoner of the whole system. As the core of an expert system, Inference Engine handles the input and output data, which in our case is the SECS-II messages sent to and replied from the target equipment. It is essentially the rules processor, where the conditions of the rules, the *if* portions, will be examined, and the conclusions will be performed, once the conditions are satisfied.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Simply speaking, during the discovery process, the program firstly sends a list of predefined SECS-II messages to the target equipment. Then based on the data in the Equipment DataBase, the Inference Engine applies the existing rules on the replies from the equipment and generates the next list of SECS-II messages to send. Finally the Inference Engine derives results from all the information it has and update the Equipment DataBase or Rule Base if necessary. Therefore, the roles of the Inference Engine are actually the data processor, inference handler, and decision maker.

The following diagram shows a basic flow chart of the Inference Engine.



ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Figure 3.3 Program flow of the Inference Engine3.2 System Implementation Overview

Referring to Figure 2.1 (*block diagram design of the proposed system*), the proposed system comprises two basic functional modules, which are the Message Discovery and the Code Generator. However, as explained earlier, the Message Discovery contains the main intelligence and the most challenging research difficulties, while the Code Generator involves complicated and tedious coding process. Therefore, this project will have the high level design of both modules, but only focuses on implementations of the Message Discovery module and its sub-modules.

In the rest of this chapter, an overview of implementation of the Message Discovery System will be given. The system is implemented layer by layer, and the implementation will be explained mainly in the form of UML implementations.

3.2.1 UML Design of the Overall System

Before going into the development of individual modules, an overall design of the whole system is needed, in order to get a general idea on the implementations of different modules. An overall design also helps in making clear the system structure and how to interface with each of the modules.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

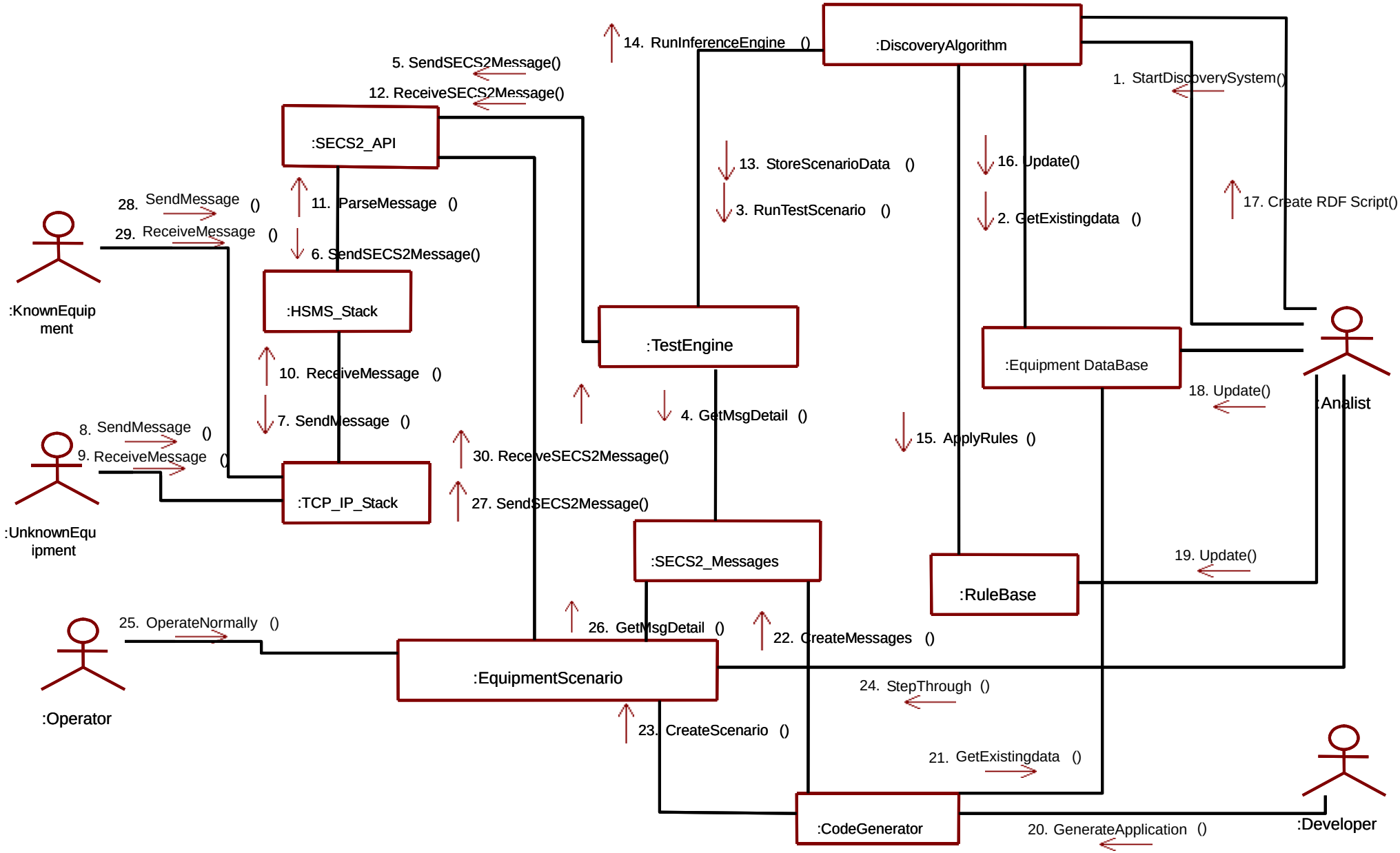


Figure 3.4 UML design of the overall system

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

As an overall system design, the UML design with the basic interfacing between modules is showed in the above diagram.

In order to have a better understanding of the system structure, and to provide easier approaches for the development, the design of the Message Discovery System is separated into three layers. They are HSMS Layer, SECS-II Layer, and Application Layer. Referring to Figure 3.1, the modules in the HSMS Layer are: HSMS-SS Stack and TCP/IP Stack, the SECS-II API is in the SECS-II Layer, and modules in the Application Layer are TestEngine, DiscoveryAlgorithm, Equipment DataBase, and Rule Base.

The whole system is implemented mainly in JAVA and C++. In order to make use of the Jena reasoning engine (in a open source JAVA package), the DiscoveryAlgorithm is developed in JAVA language. However, for a high portability and modularity of the software program, the rest of the modules in the system are developed into Dynamic Linking Libraries (DLL) using C++ language. These DLL modules will be linked by the DiscoveryAlgorithm using JAVA Native Interface (JNI). The JNI enables a program written in JAVA to easily link and call functions in a DLL that are written in C++.

3.2.2 Implementations of the Program Layers

In this section, the system implementations are shown in three steps, corresponding to the three program layers (HSMS, SECS-II, and the Application Layer).

3.2.2.1 HSMS Layer

There are two program modules in this layer, which are the HSMS-SS Stack and the TCP/IP Stack. For the TCP/IP Stack, all the necessary work has been done by *Microsoft* and has been packaged into a standard library (by the name of winsock) provided by the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

programming software (Visual Studio). Therefore in the HSMS Layer, only the HSMS-SS Stack module need to be developed. The UML implementation of the HSMS-SS Stack is described below.

By using the multi-threading technique, the HSMS-SS Stack is actually running three different tasks at the same time. At the moment when the HSMS-SS Stack is called, it generates two more program threads. These two threads are derived from a thread class, as presented in the following UML class diagram.

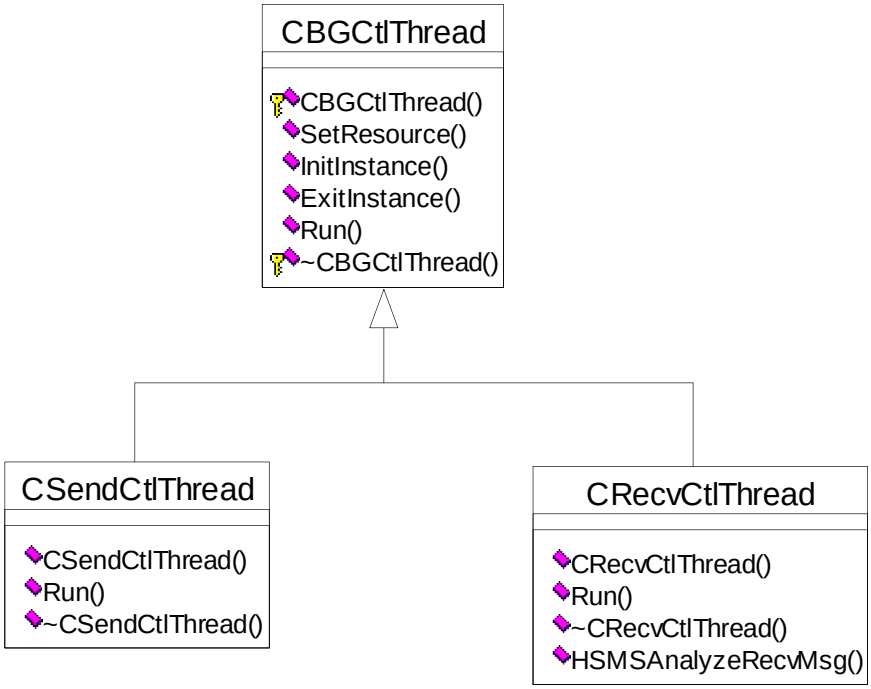


Figure 3.5 HSMS UML class diagram

CSendCtlThread, as it means from it's name, controls the sending of every SECS-II messages to the target equipment. It's a background thread that is always monitoring the communication link. When the communication link is available, it sends the SECS-II messages in the Send Message Queue in a first in first out basis. Every message will be assigned a unique message number. This message number is designed for the purpose of re-sending the message when it is timeout before getting the reply. Every out sending message, which needs a reply, will be stored in the Transaction List indexed by the message number.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

CRecvCtlThread, on the other hand, is assigned to keep monitoring the receiving end of the communication link. It captures every incoming SECS-II messages from the target equipment. Every incoming messages will be analyzed by the HSMSAnalyzeRecvMsg() function. Messages for the upper program layers will be put into the Receive Message Queue, while messages for the communication protocol will be analyzed and replied if necessary. Furthermore, before the received message is put into the Received Message Queue, it's message number will be checked. If this message is a reply of one of the previous out sending messages, the corresponding out sending message in the Transaction List will be removed to prevent it being re-send when timeout.

Other resource data classes are presented in the UML class diagram below:

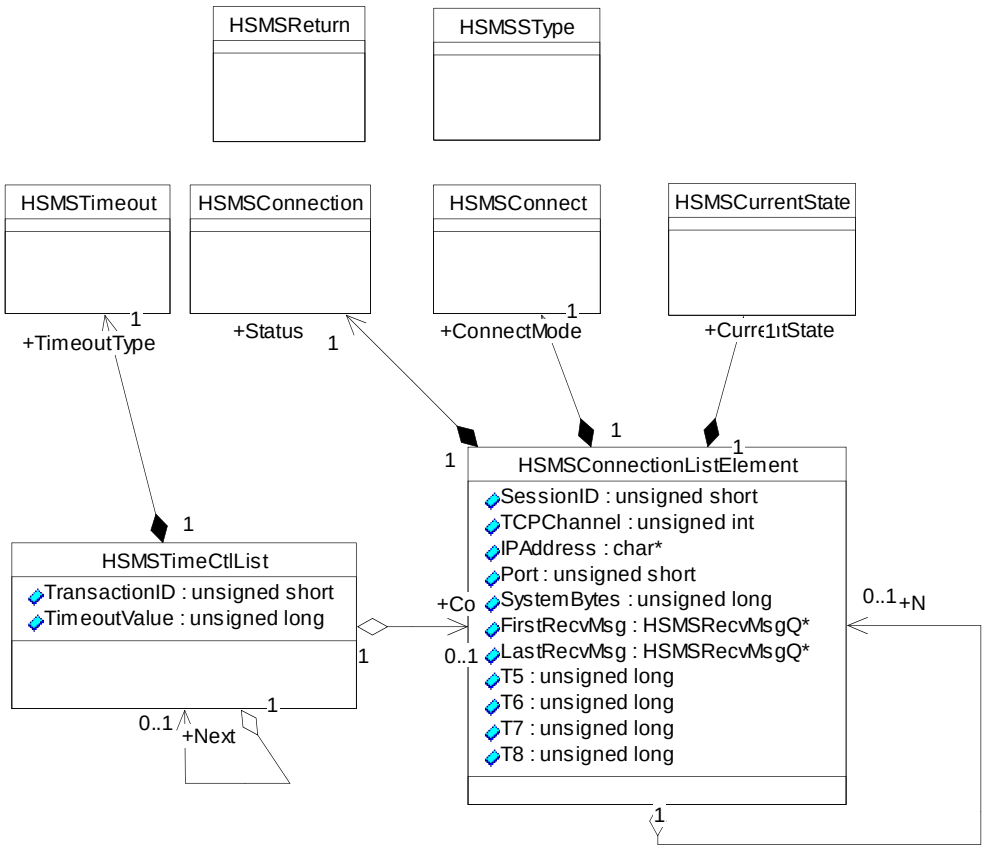


Figure 3.6 HSMS UML class diagram 2

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

3.2.2.2 SECS-II Layer

The function of the SECS-II Layer is to relay the SECS-II messages by working as a data format interpreter, for the communication between the Application Layer and the HSMS Layer. It therefore has relatively simpler structure and functionalities. The only module in this layer is SECS-II API, which is also built as a DLL library using C++ language. As an Application Program Interface (API), it is running on top of the HSMS-SS Stack, which means if the application program wants to call a function in the HSMS-SS Stack, it needs to go through the interfacing function in the SECS-II API module. By doing this, it saves the application program from doing a lot of resource allocating and data type casting.

The program interface functions of the SECS-II API and how they work will be introduced below:

- *Initialize(void)*

  This function initiates the resource data variables in the HSMS-SS Stack. It creates the two background working threads, which are the CSendCtlThread and the CRecvCtlThread.

- *CreateConnection(unsigned short SessionID, char *IPAddress, unsigned short Port)*

  This function creates a HSMS-SS connection on top of TCP/IP, to the target semicon equipment specified by the IP address. It then allocates memories to the connection variables if the connection was established successfully.

- *CloseConnection()*

  This function closes the HSMS-SS connection by terminating the two background working threads and de-allocating the related resource memories.

- *SendMessages(MsgContent*)*

  This function sends out the list of SECS-II messages that was requested by the upper program layer. By putting the list of out sending messages to the Send Message Queue

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

in the HSMS-SS Stack, this API function virtually sends out the messages to the equipment IP address specified.

- *RecvMessage(MsgContent*)*

  This function on the other hand receives the incoming SECS-II messages by retrieving them from the Receive Message Queue in the HSMS-SS Stack. It passes the incoming messages to the upper program layer by returning the pointer of the message list.

- *CheckRecvStatus(void)*

  This is actually a polling function used by the upper program layer. By making use of this polling function from time to time, the upper program layer knows whether there's any incoming message arrived in the Receive Message Queue.

Through calling the above simple API functions, the module simplifies the interfacing between the upper program layer and the communication linking layer. It hides all the detail processes of establishing connections, translating SECS-II messages, and allocating resources.

3.2.2.3 Application Layer

There are four main functional modules in this program layer. They are the TestEngine, DiscoveryAlgorithm, Equipment DataBase, and the Rule Base. Different from the modules mentioned in the above sections, the DiscoveryAlgorithm is built in JAVA language, while the TestEngine is developed into a DLL library with the JNI (JAVA Native Interface) technology. The Equipment DataBase is setup using the MySQL 4.1 database software, and the Rule Base is designed into ASCII text files for the ease of user editing. The implementations and the different features of these four modules will be given below:

TestEngine

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Similar to the SECS-II API module, the TestEngine is also built into a DLL library as an interface to the SECS-II Layer. However, the TestEngine has different functionalities than the SECS-II API. It is responsible to reconstruct the SECS-II messages in the Scenario List sent by the DiscoveryAlgorithm. It then sends out the reconstructed messages through the lower layers. Below are the program interface functions provided in the TestEngine:

- *TEinit()*

  This function is called to initialize the resources and establish a TCP/IP link to target equipment through the lower layers.

- *TEsendslist()*

  By calling this function the DiscoveryAlgorithm passes a Scenario List to the TestEngine. The latter reconstruct the messages in the list into full SECS-II messages, and pass them to the lower layers to send them out.

- *TEpoll()*

  By calling this function from time to time, the DiscoveryAlgorithm is able to retrieve the SECS-II messages once they have arrived at the TCP/IP port.

- *TEcloseconn()*

  This function stops the modules in the lower layers from running, by closing the TCP/IP connections and terminating the working threads.

DiscoveryAlgorithm

The DiscoveryAlgorithm is developed in JAVA language, and interface with the TestEngine using JAVA Native Interface technique. Compare to the rest of the program modules, this module is more important to the system and more complicated. More functions calling and more libraries interfacing are involved. The relationships and process flow of the different functions are presented in UML Collaboration Diagram

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

(Figure 3.7) and Sequence Diagram (Figure 3.8).

As the heart of the DiscoveryAlgorithm, the Inference Engine class is mainly composed of the Jena inference subsystem (Refer to Appendix E for introduction of Jena). The Inference Engine uploads the rules from the Rule Base, and then makes reasoning inference on the existing data. The derived results will then be analyzed by the DMHandler, and a new Scenario List will be generated or conclusions will be drawn.

Equipment DataBase

The Equipment DataBase is implemented in MySQL database, and the information is classified by equipment and equipment categories. This information is provided to the DiscoveryAlgorithm for the inference process. The graphical design of this relational database is given in Figure 3.9.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

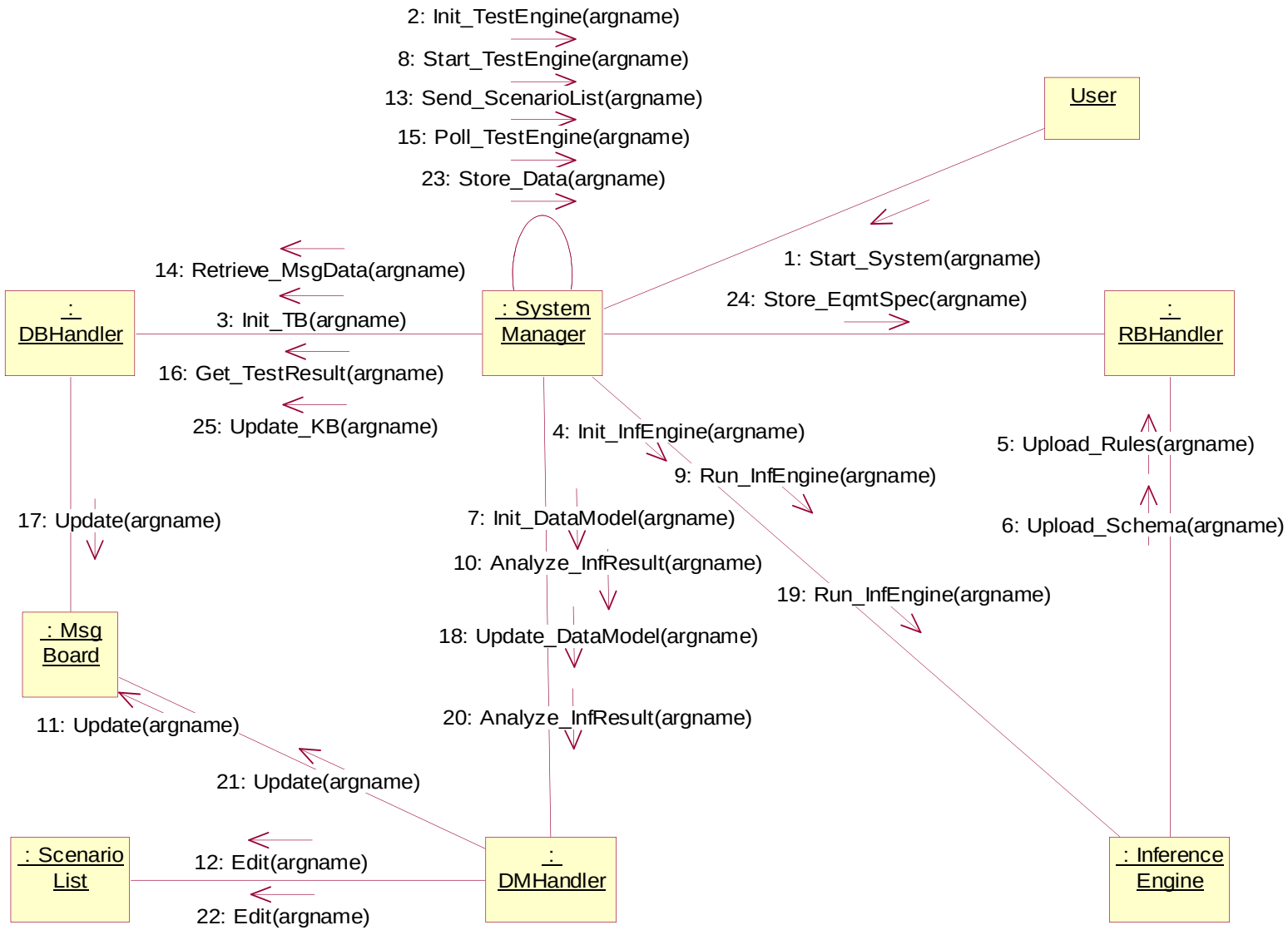


Figure 3.7 UML Collaboration Diagram of the Discovery diagram

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

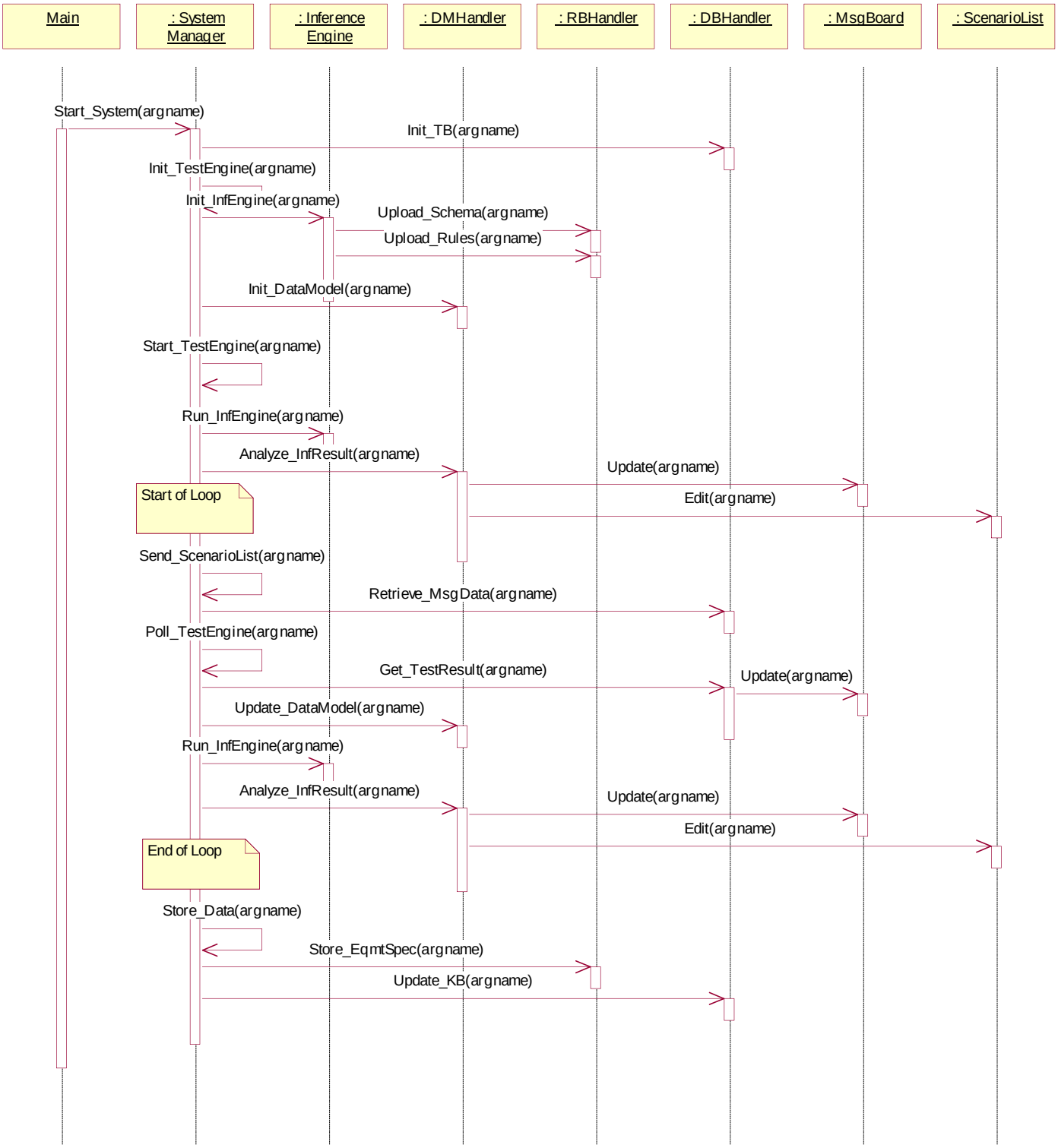


Figure 3.8 UML Sequence Diagram of the Discovery diagram

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

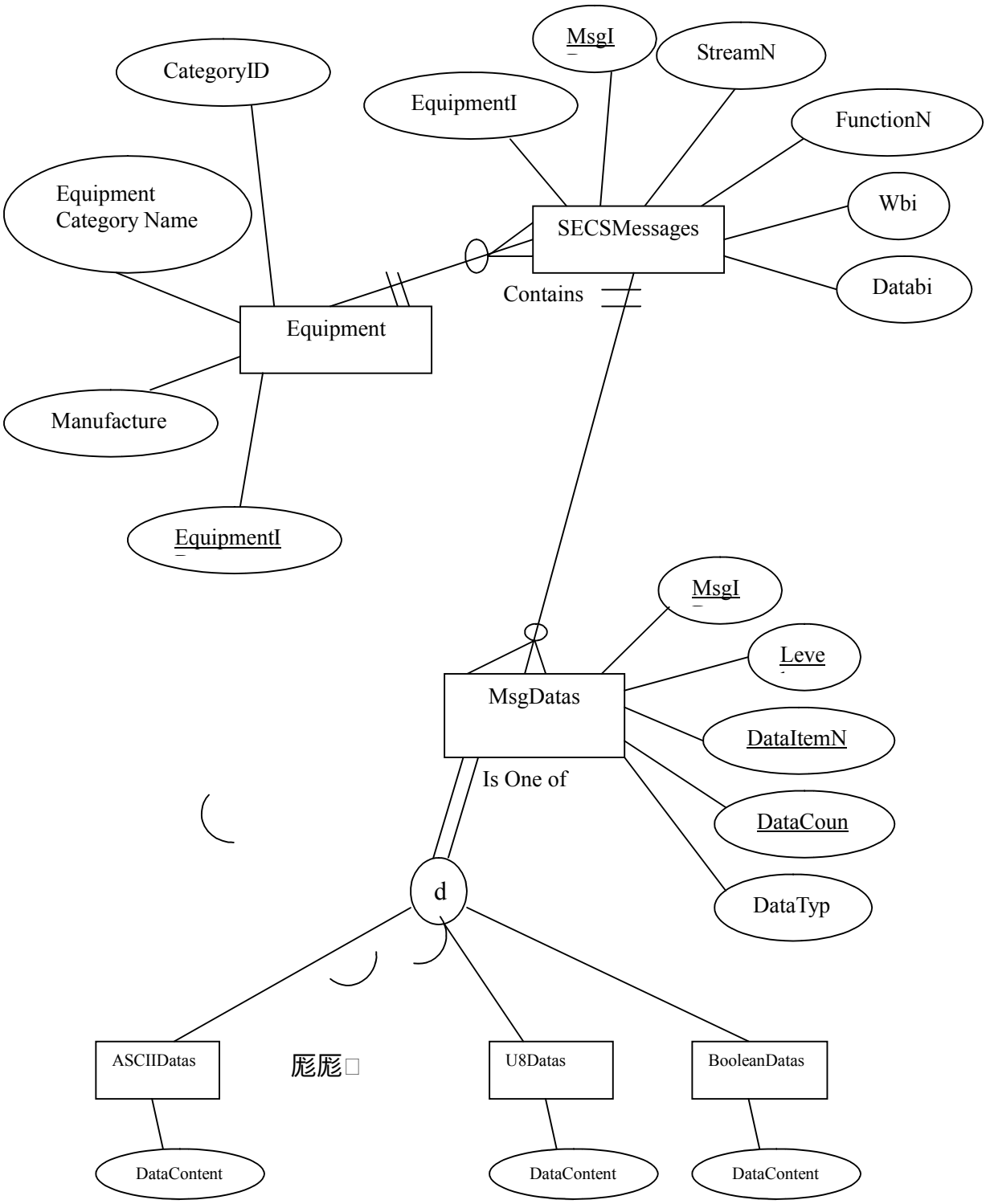


Figure 3.9 Relational Database Design of the Equipment DataBase

Rule Base

The Rule Base is implemented in ASCII text files, which contain all the rules created in RDF/XML format. Examples of the actual rules in the Rule Base are given below:

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
@prefix eq: <http://SECSMsgDiscovery/eqmt/#>
#Stream1 rules
[(?e eq:HasStdSECSMsg eq:S1F15) -> (?e eq:HasStdSECSMsg eq:S1F17)]
#Stream2 rules
[(?e eq:HasStdSECSMsg eq:S2F1) -> (?e eq:HasStdSECSMsg eq:S2F3)]
```

In the above rules examples, **`@prefix`** defines a prefix **eq** which can be used in the rules, and this prefix is local to the rule file. Because RDF is originally designed for data processing in web applications, so objects in the RDF rules are usually identified using URLs (Uniform Resource Locator), such as **`http://SECSMsgDiscovery/eqmt/#HasStdSECSMsg`**, where the front part of this object can be replaced by the prefix **`eq:`**, hence it becomes **`eq:HasStdSECSMsg`**. **`?e`** represents a variable in the rule, where **`?`** is the indicator of a variable, and **`e`** is just an arbitrary name for the variable. Any line starts with **`#`** is a comment line.

A RDF rule is basically consisting of a condition part, and a conclusion part. The condition part is the portion before the **`->`** symbol, and the conclusion part is the portion behind. In the condition part, you can have as many condition terms as possible. A condition term has three elements, which are the object, the property, and the value, and they are enclosed by a pair of brackets. In the conclusion part, you can also have as many conclusion terms as possible, and they are having the same structure as the condition terms. Condition terms in the condition part or conclusion terms in the conclusion part can be separated by commas. When a rule is applied, if all the condition terms of the rule are satisfied, the conclusion part of this rule will be fired, and the conclusion terms will be implemented.

In the above example, **`?e`** represents a variable, which in this case equals any object. So the first rule actually means if any object has the property **`eq:HasStdSECSMsg`** whose value is **`eq:S1F15`**, then this object will possess the property **`eq:HasStdSECSMsg`** whose value is **`eq:S1F17`**. The second rule has the similar meaning except the property value changed to **`eq:S2F1`** and **`eq:S2F3`**.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

More details about how and where to use rules in the Rule Base are illustrated in the Discovery Process at Section 2.3.4, and how an inference is made by applying these rules is also described. What's more, there are more syntaxes and structures for general uses of the RDF rules. If you are interested in knowing more about the inference support, you can browse the website http://jena.sourceforge.net/inference/index.html for more information.

3.3 Project Test Results

3.3.1 Software Tools

Semiconductor manufacturing equipment are expensive. Furthermore, except for maintenance, they are scheduled to operate all the time from the day they are installed. Therefore, it is hardly possible for a student to have a chance to run an under-development program on real semicon equipment. Because of this, instead of using a real equipment, an equipment simulation software is useful in this case to test the developed system. The simulation software chosen is SECSIM Pro+ 3.04 from Asyst, which is a software for programmable simulation for semicon equipment.

3.3.2 Test Procedures

Below are the test results with a simple scenario from Wire Bonder equipment. In this test, the completed modules of the system, as well as a preliminary version of the Equipment DataBase, Rule Base and a basic GUI were used. 72 standard-defined SECS-II messages were used in the test scenario:

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

S1,F1 Are You There Request (R)	S1,F2 On Line Data (D)
S1,F3 Selected Equipment Status Request (SSR)	S1,F4 Selected Equipment Status Data (SSD)
S1,F11 Status Variable Namelist Request (SVNR)	S1,F12 Status Variable Namelist Reply (SVNRR)
S1,F13 Establish Communications Request (CR)	S1,F14 Establish Communications Request Acknowledge (CRA)
S1,F15 Request OFF-LINE (ROFL)	S1,F16 OFF-LINE Acknowledge (OFLA)
S1,F17 Request ON-LINE (RONL)	S1,F18 ON-LINE Acknowledge (ONLA)
S2,F13 Equipment Constant Request (ECR)	S2,F14 Equipment Constant Data (ECD)
S2,F15 New Equipment Constant Send (ECS)	S2,F16 New Equipment Constant Acknowledge (ECA)
S2,F17 Date and Time Request (DTR)	S2,F18 Date and Time Data (DTD)
S2,F23 Trace Initialize Send (TIS)	S2,F24 Trace Initialize Acknowledge (TIA)
S2,F29 Equipment Constant Namelist Request (ECNR)	S2,F30 Equipment Constant Namelist (ECN)
S2,F31 Date and Time Set Request (DTS)	S2,F32 Date and Time Set Acknowledge (DTA)
S2,F33 Define Report (DR)	S2,F34 Define Report Acknowledge (DRA)
S2,F35 Link Event Report (LER)	S2,F36 Link Event Report Acknowledge (LERA)
S2,F37 Enable/Disable Event Report (EDER)	S2,F38 Enable/Disable Event Report Acknowledge (EERA)
S2,F39 Multi-Block Inquire (MBI)	S2,F40 Multi-Block Grant (MBG)
S2,F41 Host Command Send (HCS)	S2,F42 Host Command Acknowledge (HCA)
S2,F43 Reset Spooling Streams and Functions (RSSF)	S2,F44 Reset Spooling Acknowledge (RSA)
S2,F45 Define Variable Limit Attributes (DVLA)	S2,F46 Variable Limit Attribute Acknowledge (VLAA)
S2,F47 Variable Limit Attribute Request (VLAR)	S2,F48 Variable Limit Attributes Send (VLAS)
S4,F19 Transfer Job Create (TJ)	S4,F20 Transfer Job Acknowledge (TJA)
S4,F21 Transfer Job Command (TC)	S4,F22 Transfer Command Acknowledge (TCA)
S4,F27 Handoff Ready (HR)	-
S4,F33 Handoff Verified (HV)	-
S5,F3 Enable/Disable Alarm Send (EAS)	S5,F4 Enable/Disable Alarm Acknowledge (EAA)
S5,F5 List Alarms Request (LAR)	S5,F6 List Alarms Data (LAD)
S6,F15 Event Report Request (ERR)	S6,F16 Event Report Data (ERD)
S6,F19 Individual Report Request (IRR)	S6,F20 Individual Report Data (IRD)
S6,F23 Request Spooled Data (RSD)	S6,F24 Request Spooled Data Acknowledgment Send (RSDAS)
S7,F1 Process Program Load Inquire (PPI)	S7,F2 Process Program Load Grant (PPG)
S7,F3 Process Program Send (PPS)	S7,F4 Process Program Acknowledge (PPA)
S7,F5 Process Program Request (PPR)	S7,F6 Process Program Data (PPD)
S7,F17 Delete Process Program Send (DPS)	S7,F18 Delete Process Program Acknowledge (DPA)
S7,F19 Current EPPD Request (RER)	S7,F20 Current EPPD Data (RED)
S10,F3 Terminal Display, Single (VTN)	S10,F4 Terminal Display, Single Acknowledge (VTA)
S10,F5 Terminal Display, Multi-Block (VTN)	S10,F6 Terminal Display, Multi-Block Acknowledge (VMA)

Figure 3.10 SECS-II messages defined on the test equipment

The test procedure is as follows:

- The user keys in descriptions of the (simulated) equipment at the GUI.
- According to the descriptions, the machine is located in the network.
- The TestEngine will send out the predefined SECS-II message script to the machine.
- By receiving replies from the machine, the Inference Engine applies rules on the replies and starts the discovery process.
- During the process, the SECS-II messages will be stored in the DataBase once they have been discovered.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

- The test finishes after the program finishes the discovery process and an RDF equipment report has been generated.

Test Results

As expected, all 72 SECS-II messages of the simulated equipment have been successfully discovered. Every message discovered were displayed on the Run Time Display window of the program. Figure 3.11 shows the result window while message discovery is in process.

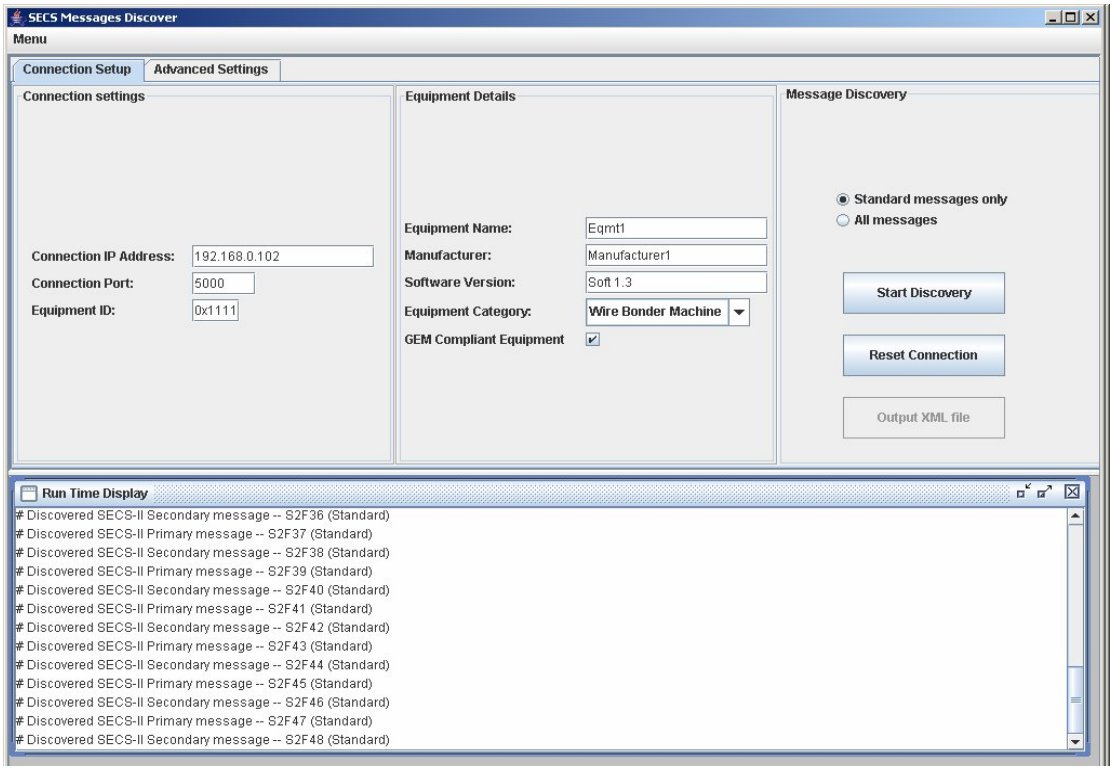


Figure 3.11 SECS-II messages being discovered

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The main aims of the tests in this phase of development are verification of the correct format of the information by sending / receiving SECS-II messages to/from the equipment and to prove that all the communication connections and discovery intelligence are working according to expectations. Finally, the test results satisfactorily proved that the proposed solution is feasible, and the project design, project implementation have been successfully completed.

Chapter 4 Conclusion and Recommendations

4.1 Conclusions

In this project, a feasible system design which could provide a semiautomatic equipment integration solution for the semiconductor industry has been developed and tested. This proposed system should alleviate the current problem faced by equipment integrators, who need to spend tedious effort to study the equipment and configure the host controller in order to integrate the equipment into a factory network. With the assistance of this system, semiconductor equipment manufacturers could deliver their newly developed equipment more quickly and the equipment end-users to start integrating the “unknown” equipment to factory network at a much earlier and easier stage.

This system is developed based on the AI technique of the Expert System. The system makes use of the knowledge gained from the SEMI communication standards to solve the target problem. The Goal-Driven Reasoning Methodology is applied in the search of equipment supported SECS-II messages, to allow the search to be carried out more efficiently. A framework for encoding knowledge (equipment data as well as rules from various standards, equipment suppliers and equipment users) into the proposed system has been established for an easier upgrade and enhancement of the system capabilities. Finally, a real scenario test had proved that the proposed solution is feasible, and the project design, project implementation have been successfully completed.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

However, there are also some limitations for the proposed system at the current stage of development. As a pilot project, the development focused on design of the solution and implementation of the methodology. At this stage, the system is lack of comprehensive knowledge of the domain, such as large amount of real equipment data and full set of rules extracted from the SEMI standards, etc. Therefore some recommendations for the further development are given below.

4.2 Recommendations

The recommendations on the future work are:

- With the framework established in this project, a more comprehensive set of rules and databases could be built up as more knowledge can be injected and more equipment are being tested with the proposed system.

- The Code Generator module of the proposed system could be developed to achieve a fully plug-and-play equipment integration. The Code Generator should automatically generate the correct interfacing program for the factory host controller to be able to control the equipment discovered by the Message Discovery module.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Bibliography

[1] SEMI URL: http://www..semi.org

[2] SIEMENS URL:

http://www.automation.siemens.com/semiconductor/html_76/inf/inf3.htm

[3] SEMI International Standards, SEMI Equipment Communications Standard (SECS): SEMI E4-91 (Message transfer - SECS I); SEMI North America International Headquarters, Mountain View CA, 2002

[4] SEMI International Standards, SEMI Equipment Communications Standard (SECS): SEMI E5-0996 (Message Content - SECS II); SEMI North America International Headquarters, Mountain View CA, 2002

[5] SEMI International Standards, High Speed SECS Message Services General Session (HSMS-GS): SEMI E37.2; SEMI North America International Headquarters, Mountain View CA, 2002

[6] SEMI International Standards, Generic Equipment Model (GEM): SEMI E30-93; SEMI North America International Headquarters, Mountain View CA, 2002

[7] A. Dugenske, A. Fraser, T. Nguyen and R.Voitus, The National Electronics Manufacturing Initiative (NEMI) plug and play factory project. *International Journal of Computer Integrated Manufacturing*. Vol. 13 No.3 (2000)

[8] David Lammers, Control software seen as 300-mm production hurdle, *EETimes* July 14, 2000 (12:28 PM EDT)

[9] SEMI Provisional Standard For The Object-Based Equipment Model (OBEM): SEMI E98-1102; SEMI North America International Headquarters, Mountain View CA, 2002

[10] George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Fourth Edition, *Pearson Education Limited*, 2002

[11] OMG URL: www.omg.org/uml

[12] UML Tutorial:
http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/diagrams.htm

[13] W3C URL: http://www.w3.org

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[14] SourceForge Jena URL: http://jena.sourceforge.net

[15] BrainMaker URL: http://www.calsci.com/Stock.html

[16] Data mining at Wikipedia URL: http://en.wikipedia.org/wiki/Data_mining

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Appendix A

A sample of the Equipment Schema:

```
<?xml version="1.0" ?>
<!DOCTYPE rdf:RDF[
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]
>

<rdf:RDF xmlns:eqmt="http://SECSMsgDiscovery/eqmt/#"
        xml:base="http://SECSMsgDiscovery/eqmt/"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">


<owl:Class rdf:ID="Equipment">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#EqmtID"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#HasStdSECSMsg"/>
      <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">756</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#HasDefSECSMsg"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="SECSMsg">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#StreamNo"/>
      <rdfs:range rdf:resource="&xsd;positiveInteger"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#FunctionNo"/>
      <rdfs:range rdf:resource="&xsd;nonNegativeInteger"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#CertaintyFactor"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:DatatypeProperty rdf:ID="EqmtID">
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
  <rdfs:domain rdf:resource="#Equipment" />
  <rdfs:range  rdf:resource="&xsd;string" />
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="HasStdSECSMsg">
  <rdfs:domain rdf:resource="#Equipment" />
  <rdfs:range  rdf:resource="#SECSMsg" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="HasDefSECSMsg">
  <rdfs:domain rdf:resource="#Equipment" />
  <rdfs:range  rdf:resource="#SECSMsg" />
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="StreamNo">
  <rdfs:domain rdf:resource="#SECSMsg" />
  <rdfs:range  rdf:resource="&xsd;positiveInteger"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="FunctionNo">
  <rdfs:domain rdf:resource="#SECSMsg" />
  <rdfs:range  rdf:resource="&xsd;nonNegativeInteger"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="CertaintyFactor">
  <rdfs:domain rdf:resource="#SECSMsg" />
  <rdfs:range  rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>

<eqmt:Equipment rdf:ID="Eqmt1">
  <eqmt:EqmtID>Eqmt1</eqmt:EqmtID>
</eqmt:Equipment>
<eqmt:SECSMsg rdf:ID="S4F1">
  <eqmt:StreamNo rdf:datatype="&xsd;int">4</eqmt:StreamNo>
  <eqmt:FunctionNo rdf:datatype="&xsd;int">1</eqmt:FunctionNo>
</eqmt:SECSMsg>
<eqmt:SECSMsg rdf:ID="S4F3">
  <eqmt:StreamNo rdf:datatype="&xsd;int">4</eqmt:StreamNo>
  <eqmt:FunctionNo rdf:datatype="&xsd;int">3</eqmt:FunctionNo>
</eqmt:SECSMsg>
<eqmt:SECSMsg rdf:ID="S7F1">
  <eqmt:StreamNo rdf:datatype="&xsd;int">7</eqmt:StreamNo>
  <eqmt:FunctionNo rdf:datatype="&xsd;int">1</eqmt:FunctionNo>
</eqmt:SECSMsg>
<eqmt:SECSMsg rdf:ID="S7F3">
  <eqmt:StreamNo rdf:datatype="&xsd;int">7</eqmt:StreamNo>
  <eqmt:FunctionNo rdf:datatype="&xsd;int">3</eqmt:FunctionNo>
</eqmt:SECSMsg>
</rdf:RDF>
```

Appendix B

A simple sample of the RDF report generated based on the target equipment's information:

```
<rdf:RDF
    xmlns:rss="http://purl.org/rss/1.0/"
    xmlns:jms="http://jena.hpl.hp.com/2003/08/jms#"
    xmlns:eqmt="http://SECSMsgDiscovery/eqmt/#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:nodeID="A0">
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#Restriction"/>
    <owl:onProperty
rdf:resource="http://SECSMsgDiscovery/eqmt/#HasDefSECSMsg"/>
    <rdfs:subClassOf rdf:nodeID="A0"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#Class">
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Datatype"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#comment">
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Container"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq"/>
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/2001/XMLSchema#int">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S1F1">
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</eqmt:FunctionN
o>
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</eqmt:StreamNo>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#SECSMsg">
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:subClassOf rdf:nodeID="A1"/>
    <rdfs:subClassOf rdf:nodeID="A2"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:nodeID="A3"/>
    <rdfs:subClassOf
rdf:resource="http://SECSMsgDiscovery/eqmt/#SECSMsg"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S2F25">
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">25</eqmt:Function
No>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</eqmt:StreamNo>
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#domain">
    <rdfs:domain
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#Eqmt1">
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S7F3"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S2F15"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S10F3"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S2F25"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S1F13"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S2F21"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S5F1"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S7F5"/>
    <rdf:type rdf:resource="http://SECSMsgDiscovery/eqmt/#Equipment"/>
    <eqmt:EqmtID>Eqmt1</eqmt:EqmtID>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S1F5"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S5F3"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S1F1"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S5F5"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S7F1"/>
    <eqmt:HasStdSECSMsg
rdf:resource="http://SECSMsgDiscovery/eqmt/#S6F3"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S10F3">
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</eqmt:FunctionN
o>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">10</eqmt:StreamNo
>
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#object">
    <rdfs:domain
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#object"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#subject">
    <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#subject"/>
    <rdfs:domain
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S7F1">
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
    <rdf:type rdf:resource="http://SECSMsgDiscovery/eqmt/#SECSMsg"/>
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</eqmt:FunctionN
o>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">7</eqmt:StreamNo>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Class">
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#range">
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:domain
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#ContainerMembershipPr
operty">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#ContainerMembershi
pProperty"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#Container">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Container"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#StreamNo">
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#positiveInteger"/>
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
rdf:resource="http://SECSMsgDiscovery/eqmt/#SECSMsg"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#isDefinedBy">
    <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#seeAlso"/>
    <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#isDefinedBy"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#List"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#Restriction">
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:nodeID="A2">
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#Restriction"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
    <owl:onProperty
rdf:resource="http://SECSMsgDiscovery/eqmt/#FunctionNo"/>
    <rdfs:subClassOf rdf:nodeID="A2"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#subClassOf">
    <rdfs:domain
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Container"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2001/XMLSchema#string">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Alt">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Container"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Alt"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S7F3">
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">90</eqmt:Certaint
yFactor>
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</eqmt:FunctionN
o>
    <rdf:type rdf:resource="http://SECSMsgDiscovery/eqmt/#SECSMsg"/>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">7</eqmt:StreamNo>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#subPropertyOf">
    <rdfs:range
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
  </rdf:Description>
  <rdf:Description rdf:nodeID="A1">
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#positiveInteger"/>
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#Restriction"/>
    <owl:onProperty
rdf:resource="http://SECSMsgDiscovery/eqmt/#StreamNo"/>
    <rdfs:subClassOf rdf:nodeID="A1"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://SECSMsgDiscovery/eqmt/#Equipment">
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    <rdfs:subClassOf rdf:nodeID="A4"/>
    <rdfs:subClassOf rdf:nodeID="A5"/>
    <rdfs:subClassOf rdf:nodeID="A0"/>
    <rdfs:subClassOf
rdf:resource="http://SECSMsgDiscovery/eqmt/#Equipment"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#Literal">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S1F5">
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</eqmt:StreamNo>
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">5</eqmt:FunctionN
o>
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
  </rdf:Description>
  <rdf:Description
rdf:about="http://SECSMsgDiscovery/eqmt/#FunctionNo">
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain
rdf:resource="http://SECSMsgDiscovery/eqmt/#SECSMsg"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
    <rdf:type
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
    <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S5F5">
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">5</eqmt:FunctionN
o>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">5</eqmt:StreamNo>
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
  </rdf:Description>
  <rdf:Description rdf:nodeID="A5">
    <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">75
6</owl:maxCardinality>
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#Restriction"/>
    <owl:onProperty
rdf:resource="http://SECSMsgDiscovery/eqmt/#HasStdSECSMsg"/>
    <rdfs:subClassOf rdf:nodeID="A5"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S6F3">
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">6</eqmt:StreamNo>
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</eqmt:FunctionN
o>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#Resource">
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://SECSMsgDiscovery/eqmt/#HasStdSECSMsg">
    <rdf:type
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:range rdf:resource="http://SECSMsgDiscovery/eqmt/#SECSMsg"/>
    <rdfs:domain
rdf:resource="http://SECSMsgDiscovery/eqmt/#Equipment"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#ObjectProperty">
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://SECSMsgDiscovery/eqmt/#CertaintyFactor">
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain
rdf:resource="http://SECSMsgDiscovery/eqmt/#SECSMsg"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S1F13">
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">13</eqmt:Function
No>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</eqmt:StreamNo>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement">
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S4F1">
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</eqmt:FunctionN
o>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">4</eqmt:StreamNo>
    <rdf:type rdf:resource="http://SECSMsgDiscovery/eqmt/#SECSMsg"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#List">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#List"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#Datatype">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Datatype"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S7F5">
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">5</eqmt:FunctionN
o>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">7</eqmt:StreamNo>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#label">
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://SECSMsgDiscovery/eqmt/#HasDefSECSMsg">
    <rdfs:range rdf:resource="http://SECSMsgDiscovery/eqmt/#SECSMsg"/>
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:domain
rdf:resource="http://SECSMsgDiscovery/eqmt/#Equipment"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:nodeID="A3">
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
    <owl:onProperty
rdf:resource="http://SECSMsgDiscovery/eqmt/#CertaintyFactor"/>
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#Restriction"/>
    <rdfs:subClassOf rdf:nodeID="A3"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S2F15">
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">15</eqmt:Function
No>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</eqmt:StreamNo>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S5F1">
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</eqmt:FunctionN
o>
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">5</eqmt:StreamNo>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S5F3">
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">5</eqmt:StreamNo>
    <eqmt:CertaintyFactor
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</eqmt:FunctionN
o>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#EqmtID">
    <rdfs:domain
rdf:resource="http://SECSMsgDiscovery/eqmt/#Equipment"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#DatatypeProperty">
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S2F21">
    <eqmt:CertaintyFactor
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</eqmt:Certain
tyFactor>
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">21</eqmt:Function
No>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</eqmt:StreamNo>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2001/XMLSchema#positiveInteger">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2001/XMLSchema#positiveInteger"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#first">
    <rdfs:domain
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#List"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#first"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#rest">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#List"/>
    <rdfs:range
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#List"/>
    <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#rest"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://SECSMsgDiscovery/eqmt/#S4F3">
    <rdf:type rdf:resource="http://SECSMsgDiscovery/eqmt/#SECSMsg"/>
    <eqmt:FunctionNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</eqmt:FunctionN
o>
    <eqmt:StreamNo
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">4</eqmt:StreamNo>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.w3.org/2000/01/rdf-schema#seeAlso">
    <rdfs:subPropertyOf
rdf:resource="http://www.w3.org/2000/01/rdf-schema#seeAlso"/>
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
  <rdf:Description rdf:nodeID="A4">
```

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

```
    <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#Restriction"/>
    <owl:onProperty
rdf:resource="http://SECSMsgDiscovery/eqmt/#EqmtID"/>
    <rdfs:subClassOf rdf:nodeID="A4"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>
```

</rdf:RDF>

Appendix C

C.1 What Is UML

Unified Modeling Language (UML) is an industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems standardized by the Object Management Group. UML simplifies the complex process of software design by using “blueprints” for software construction. Widespread adoption of UML is one of the forces contributing to developer demand for tools that can represent more intentional problem-related information [11].

C.2 Why Use UML

Modeling is the designing of software applications before coding. Modeling is an Essential Part of large software projects, and helpful to medium and even small projects as well. A model plays the analogous role in software development that blueprints and other plans (site maps, elevations, physical models) play in the building of a skyscraper. Using a model, those responsible for a software development project's success can assure themselves that business functionality is complete and correct, end-user needs are met, and program design supports requirements for scalability, robustness, security, extendibility, and other characteristics, before implementation in code renders changes difficult and expensive to make. Surveys show that large software projects have a huge probability of failure - in fact, it's more likely that a large software application will fail to meet all of its requirements on time

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

and on budget than that it will succeed. If you're running one of these projects, you need to do all you can to increase the odds for success, and modeling is the only way to visualize your design and check it against requirements before you start to code [11].

More importantly, models help us by letting us work at a higher level of abstraction. A model may do this by hiding or masking details, bringing out the big picture, or by focusing on different aspects of the prototype. In this project, UML designs of different software modules are implemented during the process of system design.

C.3 How UML Works

UML designs are mainly in the form of UML diagrams. There are several basic types of UML diagrams. Each UML diagram is designed to let developers and software users view a software system from a different perspective and in varying degrees of abstraction. Basic UML diagrams that commonly created in visual modeling tools include: [12]

Use Case Diagram displays the relationship among actors and use cases.

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

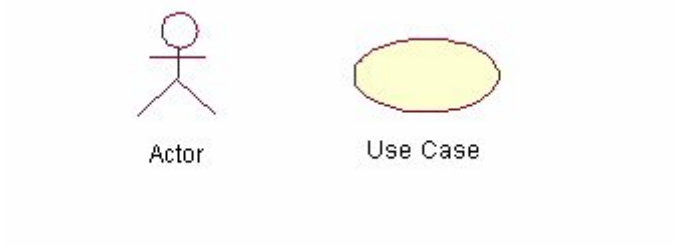


Figure C.1 Use case diagram

An actor is represents a user or another system that will interact with the system you are modeling. A use case is an external view of the system that represents some

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

action the user might perform in order to complete a task.

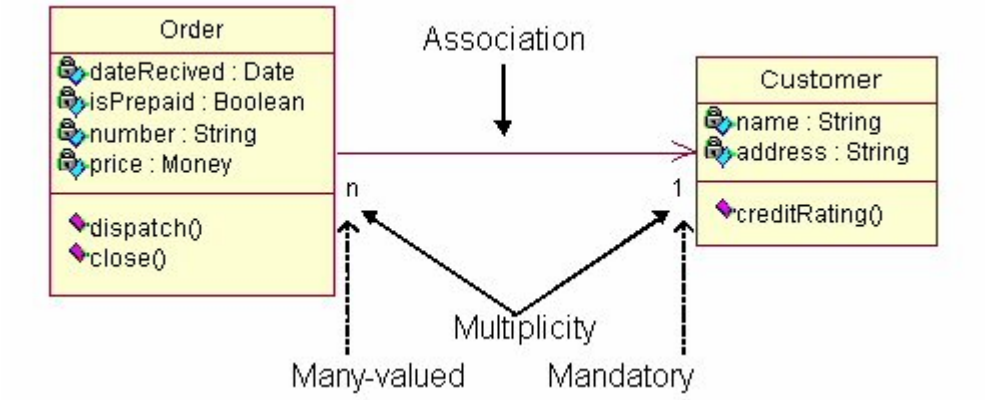
Use cases are used in almost every project. They are helpful in exposing requirements and planning the project. During the initial stage of a project most use cases should be defined, but as the project continues more might become visible.

Class Diagram models class structure and contents using design elements such as classes, packages and objects. It also displays relationships such as containment, inheritance, associations and others.

Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design. Classes are composed of three things: a name, attributes, and operations. Below is an example of a class.

Figure C.2 Class diagram

Class diagrams also display relationships such as containment, inheritance, associations and others. Below is an example of an associative relationship:



ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Figure C.3 Associative relationship

Interaction Diagrams model the behavior of use cases by describing the way groups of objects interact to complete the task. The two kinds of interaction diagrams are sequence and collaboration diagrams.

Sequence Diagram displays the time sequence of the objects participating in the interaction. This consists of the vertical dimension (time) and horizontal dimension (different objects).

Sequence diagrams demonstrate the behavior of objects in a use case by describing the objects and the messages they pass. The diagrams are read left to right and descending. The example below shows an object of class 1 start the behavior by sending a message to an object of class 2. Messages pass between the different objects until the object of class 1 receives the final message.

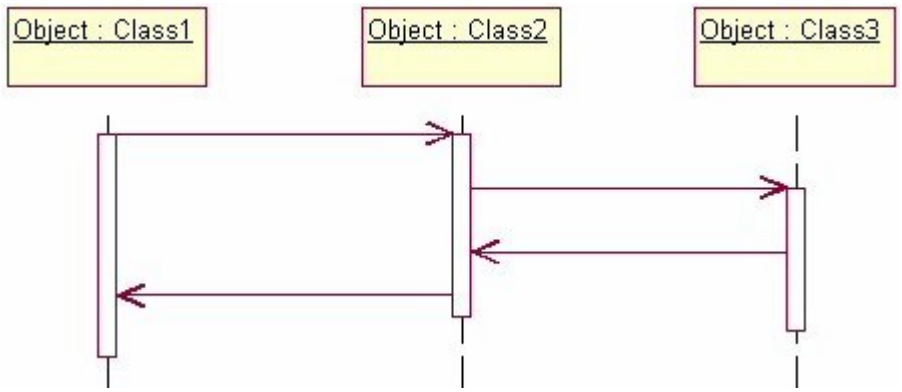


Figure C.4 Sequence diagram

Collaboration Diagram displays an interaction organized around the objects and their links to one another. Numbers are used to show the sequence of messages.

Collaboration diagrams are also relatively easy to draw. They show the relationship between objects and the order of messages passed between them. The objects are listed as icons and arrows indicate the messages being passed between them. The numbers next to the messages are called sequence numbers. As the name suggests, they show the sequence of the messages as they are passed between the objects. There are many acceptable sequence numbering schemes in UML. A simple 1, 2, 3... format can be used, as the example below shows, or for

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

more detailed and complex diagrams a 1, 1.1 ,1.2, 1.2.1... scheme can be used.

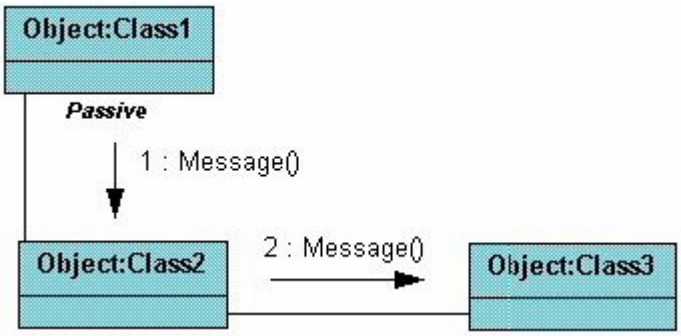


Figure C.5 Collaboration diagram

In this project, UML is mainly used in the design processes of the high level abstraction of the proposed system as well as the different modules in the system. Because it's independent of particular programming languages and development processes, designing the system in UML also provides this project the extensibility and specialization mechanisms to extend the core concepts.

Appendix D

D.1 What Is RDF

The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. It is originally intended for representing metadata about Web resources, such as the title, author, and modification date of a Web page, copyright and licensing information about a Web document, or the availability schedule for some shared resource. However, by generalizing the concept of a "Web resource", RDF can also be used to represent information about things and information that can be processed by applications, even when they cannot be directly retrieved through the Web. Examples include information about items available from on-line shopping facilities (e.g., information about specifications, prices, and availability), or the description of a Web user's preferences for information delivery.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

RDF is intended for situations in which this information needs to be processed by applications, rather than being only displayed to people. RDF provides a common framework for expressing this information so it can be exchanged between applications without loss of meaning. Since it is a common framework, application designers can leverage the availability of common RDF parsers and processing tools. The ability to exchange information between different applications means that the information may be made available to applications other than those for which it was originally created [13].

The detail RDF syntax can be found on http://www.w3.org/RDF/

D.2 Why Use RDF

As a framework to describe resources, RDF is designed to represent information in an open and flexible way. Because of its abstract and rigorously defined syntax, it can be used in isolated applications while the information formats can be easily understood by other applications. Hence the ability of easily sharing information over applications, gives RDF the great value in representing data resources among a network or even over the entire internet. This is also the main purpose of using RDF to describe the data in this project.

However, RDF is not just designed to be an easily accessible data framework. There are more reasons and motivations that make RDF suitable for this application. The followings are some of the design goals that RDF is intended to meet, and theirs suitability to be used in this project.

- **Simple Data Model**

  RDF has a simple data model that is easy for applications to process and manipulate. The data model is independent of any specific serialization syntax.

- **Formal Semantics and Inference**

  RDF has a formal semantics which provides a dependable basis for reasoning

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

about the meaning of an RDF expression. In particular, it supports rigorously defined notions of entailment which provide a basis for defining reliable rules of inference in RDF data.

- **XML-Based Syntax**

  RDF has a recommended XML serialization form, which can be used to encode the data model for exchange of information among applications.

  RDF can also use values represented according to XML schema data types, thus assisting the exchange of information between RDF and other XML applications.

- **Anyone Can Make Statements About Any Resource**

  To facilitate operation at Internet scale, RDF is an open-world framework that allows anyone to make statements about any resource.

  In general, it is not assumed that complete information about any resource is available. RDF does not prevent anyone from making assertions that are nonsensical or inconsistent with other statements, or the world as people see it. Designers of applications that use RDF should be aware of this and may design their applications to tolerate incomplete or inconsistent sources of information.

D.3 How RDF Works

RDF is based on the idea of identifying things using Web identifiers (called Uniform Resource Identifiers, or URIs), and describing resources in terms of simple properties and property values. This enables RDF to represent simple statements about resources as a graph of nodes and arcs representing the resources, and their properties and values [13]. For example, a group of statements "there is a Person identified by http://www.w3.org/People/EM/contact#me, whose name is Eric Miller, whose email address is em@w3.org, and whose title is Dr." could be represented as the RDF graph in the Figure below.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

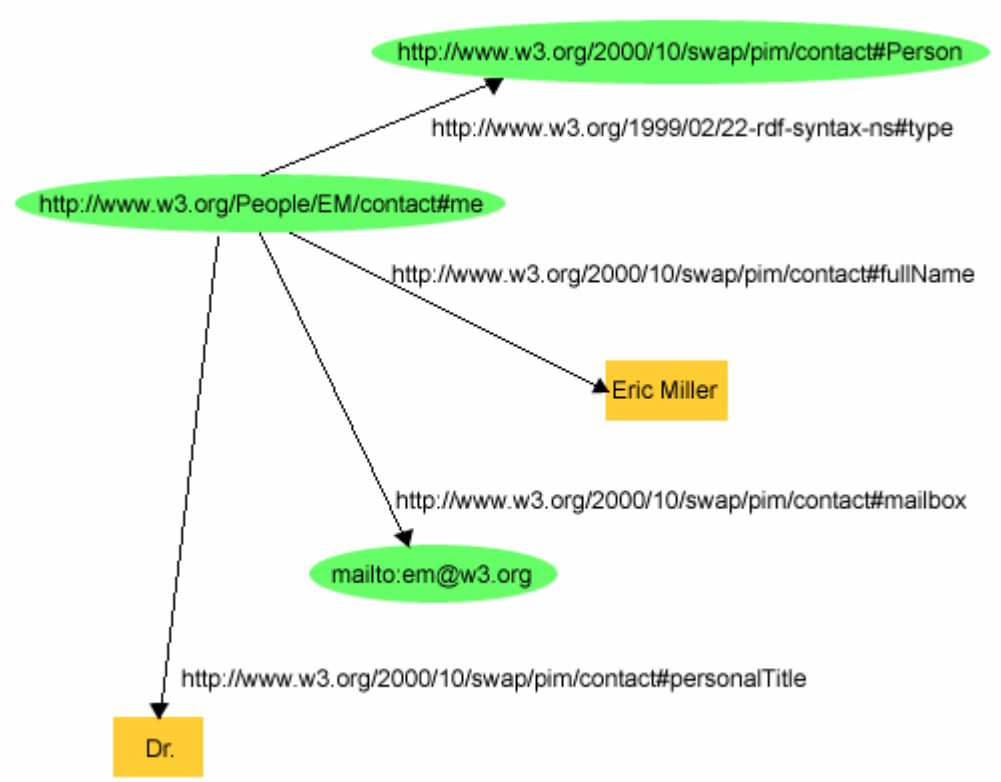


Figure D.1 URIs in a RDF graph

This example illustrates that RDF uses URIs to identify:

- individuals, e.g., Eric Miller, identified by http://www.w3.org/People/EM/contact#me
- kinds of things, e.g., Person, identified by http://www.w3.org/2000/10/swap/pim/contact#Person
- properties of those things, e.g., mailbox, identified by http://www.w3.org/2000/10/swap/pim/contact#mailbox
- Values of those properties, e.g. mailto:em@w3.org as the value of the mailbox property (RDF also uses character strings such as "Eric Miller", and values from other data types such as integers and dates, as the values of properties).

RDF also provides an XML-based syntax (called RDF/XML) for recording and exchanging these graphs. This feature greatly increases the capability of sharing

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

information among applications. Applications do not need to have a special parser or interpreter to understand the resources described by other applications. All they need is the ability to read XML, which is the widely known and adopted data description language. Below is a small chunk of RDF in RDF/XML corresponding to the above example.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                    xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>

</rdf:RDF>
```

Like HTML, this RDF/XML is machine processable and, using URIs, can link pieces of information across the Web. However, unlike conventional hypertext, RDF URIs can refer to any identifiable thing, including things that may not be directly retrievable on the Web (such as the person Eric Miller). The result is that in addition to describing such things as Web pages, RDF can also describe cars, businesses, people, news events, etc. In addition, RDF properties themselves have URIs, to precisely identify the relationships that exist between the linked items.

D.4 About OWL

OWL stands for Web Ontology Language, it is intended to be used when the information contained in documents needs to be processed by applications. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called ontology [13].

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

OWL is built on top of RDF, and it is designed to be interpreted by computers. OWL and RDF are much of the same thing, but OWL is a stronger language than RDF, not only with greater machine interpretability, but also with a larger vocabulary and stronger syntax. RDF can formally describe the exact meaning of the data, but if reasoning tasks are expected to be performed on the data, the more powerful language OWL is needed.

Appendix E

E.1 What Is Jena

Jena is a JAVA framework developed for building semantic web application. It provides programmatic environments for RDF and OWL [14]. Jena has a Java API which can be used to create and manipulate RDF graphs, and more importantly, it has a rule-based inference engine, which supports the inference process of the data represented using RDF/OWL.

The Jena inference subsystem is designed to allow a range of inference engines or reasoners to be plugged into Jena. Such engines are used to derive additional RDF assertions which are entailed from some base RDF together with any optional ontology information and rules associated with the reasoner. The primary use of this mechanism is to support the use of languages such as RDFS and OWL which allow additional facts to be inferred from instance data and class descriptions. Generally speaking, the machinery includes a generic rule engine that can be used for many RDF processing or transformation tasks.

E.2 Why Use Jena

Referring to the previous chapter, the proposed system contains an Expert System as the core of the DiscoveryAlgorithm. Based on asking questions and getting replies

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

from the target equipment, the Expert System derives and draws results from the existing information. Therefore, a rule-based inference engine is crucial for developing the Expert System. Since the Jena inference engine meets all we need and has an easy-to-use JAVA interface, the proposed Expert System is decided to be developed in JAVA using the Jena inference engine.

E.3 How Jena Works

Jena is a Java API which can be used to create and manipulate RDF graphs, and it has object classes to represent graphs, resources, properties and literals. The structure of resources, properties and literals is called a model and represented by the Model interface [14].

The following diagram provides the basic ideas of how the Jena inference engine works:

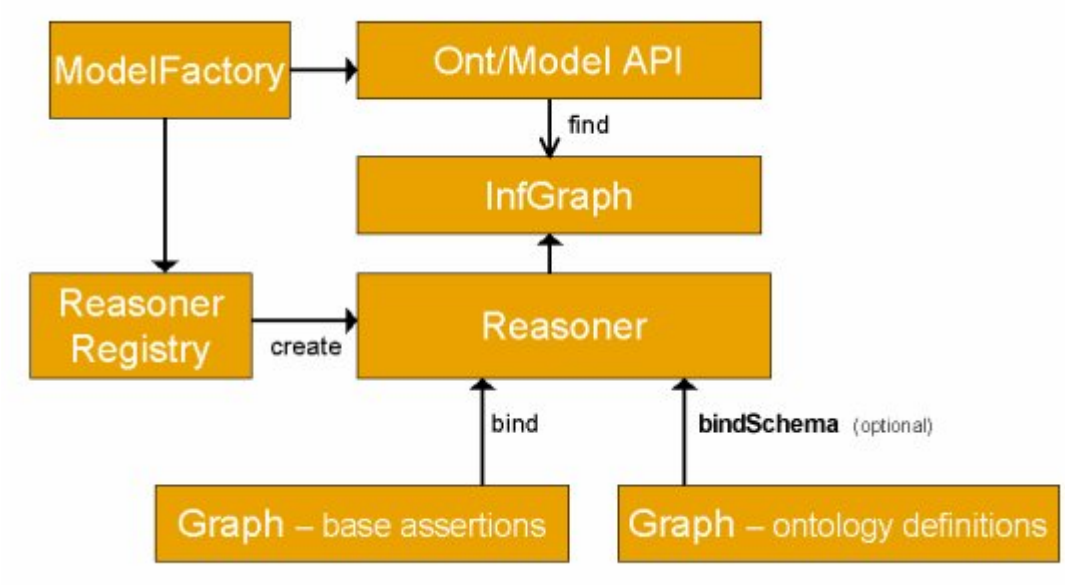


Figure E.1 Jena inference engine

By running the inference engine located in the Reasoner, additional information can be derived by the InfGraph based on the existing data stored in the Graphs. This additional information then can be added to an old model or used to create a new model by the Ont/Model API.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library