REUSABILITY TECHNOLOGY IN DEVELOPING EXPERT MODULE FOR ITS USING QPT APPROACH

ROHANI BINTI ABU BAKAR

FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2004

REUSABILITY TECHNOLOGY IN DEVELOPING EXPERT MODULE FOR ITS USING QPT APPROACH

ROHANI BINTI ABU BAKAR

DISSERTATION SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR DEGREE OF MASTER OF SOFTWARE ENGINEERING

FACULTY OF COMPUTER SCIENCE AND INFORMATION SYSTEM
UNIVERSITY OF MALAYA
KUALA LUMPUR

NOVEMBER 2004

A B S T R A K

Sistem Pembelajaran Pintar (Intelligent Tutoring System-ITS) telah menjadi satu alat bantuan mengajar yang penting masakini terutamanya dalam persekitaran pembelajaran-kendiri. Dewasa ini banyak ITS telah dibangunkan dengan menggunakan senibina yang hampir sama tetapi mengandungi modul pakar yang berbeza. Kebiasaannya, modul pakar ini dibangunkan didalam setiap pembangunan ITS walaupun dengan menggunakan maklumat atau pengetahuan yang sama. Didalam usaha untuk mengurangkan kos pembangunan, kemungkinan untuk pengetahuan atau maklumat dipakai semula dengan membangunkan perisian yang membenarkan pengguna membina modul pakar dan guna kembali bila diperlukan telah dikaji. Dua topik dari bidang sains yang berbeza, iaitu fizik dan kimia telah dipilih untuk dimodelkan untuk menguji keberkesanan prototaip perisian yang dibangunkan untuk membangunkan modul pakar. Pengetahuan yang dimodelkan dihasilkan dalam bentuk teks untuk menerangkan bentuk grafik; menerangkan domain subjek yang dibangunkan.

A B S T R A C T

Intelligent Tutoring System (ITS) has becomes an important teaching aid especially in self-learning environment. Many ITS that have been built have similar architecture but contains different expert module. Traditionally, the expert module was constructed in every ITS development even through the knowledge representation is reusable. In order to reduce the reconstruction cost, we study the possibility of making the knowledge representation reusable by building software that allows the compromise of the expert knowledge to be created and reused. Qualitative Process Technique has been used as the knowledge representation for modeling different subject domains. Two different science subjects; chemistry and physics have been chosen as models to test the effectiveness using a prototype that will be developed in this study to generate an expert module. The knowledge which is modeled is shown using a narrative text that explains the graphical representation of the subject domain.

Chapter One

Introduction

1.1 Background and Rationale

The effective use of computer technologies has becoming a long-standing issue in the field of education technology. Various researches have been done in this area to improve and enhance the teaching and learning methods. These include the advancement of teaching tools for delivery; presentations using multimedia effects, collaborative learning using Computer Support Cooperative Work (CSCW), automated assessment tools from optical character recognition (OCR) to automated essay grading tools [11]. Courseware is most commonly used in improving delivery method and it offers high interactivity between the learner and the system in making learning an exciting experience.

However, interactivity in many conventional courseware is limited by intelligence response to the idiosyncratic learning of a student. A mere plain response such as "/" (tick) to indicate correctness and "X" (cross) for false answer with animated sound does not deliver the truest need in the pedagogical theory. Intelligent Tutoring System (ITS) is an area of artificial intelligence in education that attempts to address issues of interactivity in a more intellectual way than the plain graphical and animated sounds. Researchers in ITS have focused on different components of ITS such as the student module, expert module, pedagogical module, communication module and explanation module.

ITS can be considered as more advanced compared to conventional Computer-Aided Learning (CAL) and Computer-Based Learning (CBL) technology in terms of knowledge and reasoning capabilities. ITS emphasizes on a learner's specific objectives, needs and preferences based on his or her capability. The symbiotic relationship between pedagogical module and expert module has made the interaction between ITS and the student resembles a real human tutor.

Building expert modules has led to different approaches and methodologies as it depends very much on the types of knowledge to be delivered, the problems that it addresses, the pedagogical theory it adopts and its representation styles. Nevertheless, we still believe that all of the differences can be synchronized into a strand of commonness of ITS features. The common features can be generalized and transformed into components of reusability such that any expert module developers can reduce significant amount of effort through utilizing the pre-existing bits. This will subsequently reduce the cost and programming effort, which are becoming the main concern in the theory of reusability of software engineering.

The weakness with current ITS is that the tool is not reusable at all. That means, a developer always builds one ITS for just only one subject. Even though the developer had spent a lot of time and money in developing an ITS, they need to repeat the similar process for different subject and different ITS.

Therefore, this study focuses on how to make ITS reusable. Several earlier studies regarding on how we can produce an ITS for different domain were explored. The main aim of this study is to develop a reusable tool for developing an expert module as one component in ITS. This tool will help experts in producing learning material more effectively.

The focus of this study however, is to make activities for building expert module as a routine task that is independent of any domain. An analogy to this concept is the PowerPoint, where many authors can produce different types of presentations for different purposes by using several available templates and backgrounds. Differences in choices and options are built as pre-existed functions, which can be tapped easily into the existing templates. In similar fashion, an expert module builder does not need to meddle with programming source code in encoding knowledge of different domains.

Besides that, the knowledge is amendable without the need to recompile the source code of the program. In order for this to happen, the technique of reusability is necessary. For this study, we have chosen template-based as the ideal technique to be applied into building expert module. The following subsections, describe the motivation, objectives, scopes and rhetorical structure of this study.

1.2 Motivation for the Study

Developing ITS is not a simple process as the structure depends on the experts of the domain, the pedagogical theory being used and programming styles which differ from domain to domain. It is reported that in order to produce an hour of ITS instruction for a single domain knowledge they need to spend hundred hours of development [5]. For this reason, it is proposed that the concept of reusability is adopted to make the building of ITS more effective and economical.

In order to reduce the time and cost in developing ITS, reusability concept can be used to support different domain knowledge. Consequently, the ITS is developed as component based so that it can be used again in different domains. This study aims to explore the opportunity of developing a tool for producing expert module in several domains.

1.3 Statement of the Problem

Since knowledge representation has becoming a big issue in developing ITS, many researchers have tried new approaches in modeling knowledge. The problem arises when researchers need to represent the knowledge in an effective way. In the conventional way of developing ITS, the domain expert will import his knowledge to the programmer in which the programmer will then write the program to present the expert's knowledge in the expert module.

This approach takes a lot of time and is not very effective because the information that needs to be delivered to students might be missing or misinterpreted by the programmer. Beside that, the cost for the ITS development is high because each domain need a separate ITS. This dissertation attempts to develop an expert module tool that can be used to develop ITS for different subject using Qualitative Process Techniques (QPT) as the common knowledge representation.

1.4 Objectives of the Study

The objective of the study is to develop an expert module with the following criteria:

1. Using new approach to build an expert module using QPT approach to represent the knowledge.
2. Using reusability concept in constructing expert module integrated with QPT so that, this module could be reused for other subjects.
3. Represent knowledge in graphical form and generate its relevant meaning in narrative text.

1.5 Research Questions

Specifically, the study aims to ensure the following research question: Can the tool that generates an expert module with the following criteria be developed for different subjects?

1. Capable to present knowledge using QPT approach
2. Apply reusability concept in constructing tool and generate expert module
3. Capable to present knowledge in graphical form and generate the relevant meaning using narrative text according to the graphical form.

In order to develop this reusable tool; two research questions are addressed in this study. Firstly, how can we generalize the expert module so that its modeling is amenable to different domain application. Secondly, how qualitative process theory can be extended to capture different types of knowledge.

1.6 Scope of the Study

For the purpose of this study, a prototype was developed and tested to be used to produce physics and chemistry expert modules for secondary school syllabus in Malaysia educational system. For this purpose, only a few topics from both subjects were chosen as a samples. An expert module was developed instead of the complete ITS. The expert module was built using QPT; as knowledge representation for this case study.

1.7 Contribution of the Study

The contributions of this study are:

1. Introducing the alternative way on representing knowledge for building an expert module as a component for an ITS by using QPT approach.
2. Compromising reusable concept in developing tools for generating expert module, thus, the expert module for several domains of knowledge can be generated from one software tool.

1.8 Structure of the Study

This study explores the questions and concepts behind the idea to build a reusable expert module using QPT technique. The report is organized into five chapters briefly discussed below.

Chapter one discusses the background of the problem, the objective and the scope of the study. Chapter two concentrates on the review of literature on Intelligent Tutoring System, knowledge representation and reusability topic. This chapter discusses the concept, architecture and basic components to develop ITS; the concept, taxonomy and reusability components generating or developing of expert module, and lastly the types of knowledge, knowledge representation and the use of QPT in knowledge representation.

. The architecture of prototype will be discussed in chapter three with an emphasis on how knowledge can be generated in narrative text from the relationships and the objects that are defined by the expert. Chapter four discusses the findings of the study. Lastly chapter five presents the conclusion of the study and some suggestions for future work in this area of research.

Chapter Two

Literature Review

2.1 Introduction

Intelligent Tutoring System has become an important teaching aid in the new era. In order to develop the ITS, experts from several fields are needed to ensure the quality of ITS developed. An expert module is the most important component compared to other components such as student module and tutoring module.

This chapter will review the related topics and researches that have been done before. The discussion will start with brief explanation about ITS before proceeding to the reusable concept and the knowledge representation. This chapter will give the basic idea and explanation on the proposed ideas that have been discussed in the chapter before.

2.2 Overview of Intelligent Tutoring System

Intelligent Tutoring System (ITS) emerges with the inspiration to substitute a human tutor with computer. IEEE has defined ITS as a learning technology system that dynamically adapts learning content to a learner's specific objectives, needs, and preferences by using its expertise in instructional methods and the subject to be taught [7]. That means, ITS is designed to reproduce key features of a human teacher's behavior. Usually, an ITS possesses knowledge about the domain, tutoring strategies and methods, and also is able to model the state of a student's knowledge.

ITS allows the emulation of the teacher in a sense that an ITS knows what to teach (domain content), how to teach it (instructional strategies), and learns certain relevant teaching information about the student being taught. This requires the representation of a domain expert's knowledge (called expert module), an instructor's knowledge (called the instructional module), and the particulars of the student being taught (called student module) [6]. ITS acts as the student's private tutor, while the trainer or teacher is free to focus on more complex and individualized student needs.

The subsequent subsections will describe the architecture of the ITS while addressing issues in general context.

2.2.1 An Architecture of ITS

There is a widespread agreement within the ITS community that an ITS consists of four standard modules, as depicted in figure 2.1 [3,10].

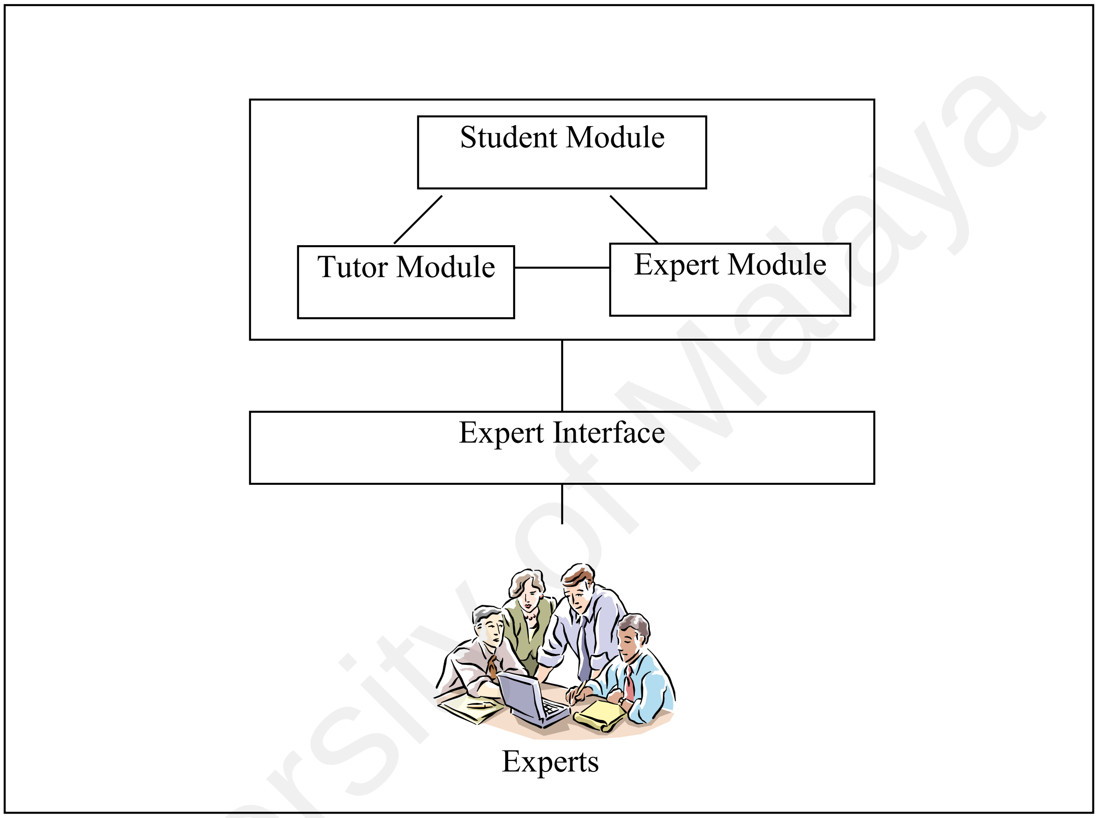


Figure 2.1:
Standard ITS Architecture

The expert module contains the domain knowledge. Expert module is also known as the knowledge base. Basically, an expert module must contain the knowledge to be delivered to the learner [5]. The tutor module identifies the deficiencies of knowledge existed in the student and selects the appropriate instructional strategies in order to present that knowledge [3]. The student module generates the framework for identifying the student's misconceptions and sub optimal performance [6]. The student module is at the heart of an ITS in that it is the component which enable individualized decision making [12]. Meanwhile, the expert interface is responsible for the communication with the learner.

Based on this standard of architecture, researchers have proposed their own additional module or component such as pedagogical module , communication module and human teacher module to make ITS more reliable, effective and expert friendly[5,9,16]. But in this report, only the expert module will be discussed later as the purposed of the study to build a reusable expert module.

2.2.2 Expert Module in ITS

An expert module contains the knowledge to be taught to the learner. It not only contains the knowledge but also skills of an expert [5]. The main function of this module is to represent knowledge possessed by an expert. Besides that, the expert module detects those concepts and contents that are not known or not used properly, making the necessary changes to the instructional plans.

Stankov defines the characteristics of ITS knowledge which is contained in an expert module as follow: (i) domain knowledge containing objects, relations among them, explanations, examples and exercises, (ii) teacher or expert's knowledge as a strategy for the process of learning and teaching and (iii) students' knowledge as a model which is dynamically generated as a result of overlying in with teachers' knowledge [13].

Although the majority of ITS packages represent knowledge in their expert module by using production rules, semantic networks have had a great impact on domain knowledge representation from the early days of ITS development already. For example, classic ITS systems such as SCHOLAR and WHY use semantic networks for representing geography knowledge, rainfall and evaporation knowledge respectively [3,4]. Meanwhile, the expert module for GUIDON is represented in three tiers, which are: (i) rules, lists, and tables, (ii) supporting information and (iii) meta level abstraction. [24]

Meanwhile Stankov has used semantic network with a frame to represent the knowledge in Tex-Sys's expert module. The basic components of Tex-Sys (Tutor Expert System) semantic network are nodes and links. Nodes are used for presenting the subject of the knowledge, while links showing relations between pairs of objects. Beside nodes and links, the system supports properties and frames (attributes and respective values), along with the property that it inherits [13].

Currently, recent researchers in this area have come up with different approaches for representing knowledge in the expert module. Dicheva suggests presenting the knowledge in expert module in two forms when she develops ITS called ITELS for terminology learning, which are implicit and explicit knowledge [17]. Dicheva tried to corporate these two forms of knowledge by using semantic knowledge as well. Thus, in addition to the linguistic knowledge, his expert model is extended to incorporate semantic knowledge and the corresponding inference engine.

An expert module becomes an important component in the ITS shell. A knowledge domain that is to be represented through tutor or tutoring module will be obtained from this module. Several researches are ongoing in finding ways to represent knowledge for effective use for ITS developers. Qualitative reasoning is a technique to model dynamic system [20]. It is more robust in terms of expressiveness and flexibility compare to conventional rule-based system. Qualitative Process Theory is one of qualitative reasoning techniques that could be adopted in developing expert module.

Usually, ITS is developed only for one domain of knowledge. For example, SCHOLAR is developed for geography knowledge, ITELS for linguistic and FLUTE for teaching formal language [21]. Thus, the production of ITS is not easy, much of the efforts, time and money are needed to spend before producing one ITS for only one domain knowledge.

Thus this study stands to prove that the reusable expert module can be developed by using QPT approach in this work. In the following chapter, we will discuss the reusable concept in developing expert module which is capable to generate different domain knowledge.

2.3 Reusability in Software Engineering

Several attempts have been made by researchers to overcome this crisis. One of the most promising proposals suggests the concept of reusability to defeat this crisis [32]. Reusability means the ability of code to be reused, in whole or in part, in new applications. Regardless of what is or when it is reused, the aim of reusing is supposed to increase the productivity, quality and reduce cost [32,33,34].

Reuse can be incorporated into the system development process by incorporating a specific reuse activity as in figure 2.2 [27] :

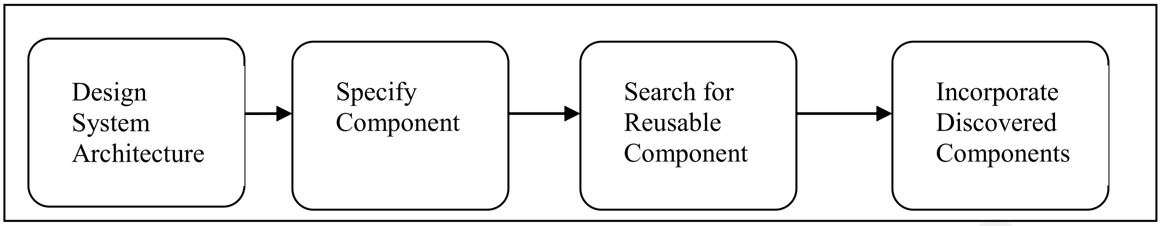


Figure 2.2 :
Reuse In a Standard Development Processes

Reuse in a standard development begins by designing system architecture of the application. In the next level, a designer will complete the high-level of design and specifications for each component. All of these components will be used in the next level to find an appropriate component to reuse it. Then, in the last stage, the designer will incorporate this component in the development to perform one new software or application.

On the other hand, when the development involves reusable component, the system requirements are modified according to the availability of the reusable component [27]. The design stage should also be based on the existing component. Then, there may be requirements to compromise in order to make this component integrated with the reusable component. The design may be less efficient than the special-purpose design.

2.3.1 Taxonomy of Reusability

Today, concepts in reuse technology have moved beyond the notions of reusability of components in libraries towards collections of more flexible design artifacts [31].

Before, researchers and software developer tried to adopt a reuse concept only in a library or code or algorithms of programming but today developers try to reuse beyond the library or algorithms, into components and applications themselves to solve different problems.

Diaz proposed a way to classify the high level taxonomy of software reusability based on conceptually independent facets and list the major factors needed to define any reuse activity [30]. The taxonomy is represented in a tuple scheme and is used as a structured vocabulary to describe a reuse task or program. Below, table 2.1 shows the taxonomy of software reuse as proposed by Diaz.

By Substance	**By Scope**	**By Mode**	**By Technique**	**By Intention**	**By Product**
Ideas, concept Artifacts, Components Procedures Skills	Vertical Horizontal	Planned, Systematic Ad-Hoc, Opportunistic	Compositional Generative	As-is, Black box With-modification, White box	Source code Design Specification Objects Text Architecture

Table 2.1:
Faceted Taxonomy of Software Reusability By Diaz [30]

The *by-substance* facet defines the essence of the items that are reused. Ideas and concepts include formal models such as the algorithms or mathematical formulas that are used to solve engineering problems. Artifacts and components refer to the source code components including subroutines, functions, and objects. Procedures and skills include processes that are used for developing software such as guidelines and methods and expert knowledge in the form of experience, observations, and recommendations.

The *by-scope* facet defines the form and extent of the reuse activity. Vertical reuse is the reuse of software within the same domain or application area. The goal of vertical reuse is to derive generic models for families of systems that can be used as standard templates for assembling new systems. Horizontal reuse is the development of generic parts that can be used in different applications such as scientific subroutine libraries and the UNIX tools.

The *by-mode* facet defines how the reuse activity is conducted. Planned reuse is the systematic and formal practice of reuse. It implies that guidelines and procedures for reuse have been defined, and that metrics are collected to assess and measure reuse performance. Ad-hoc is usually referred to as opportunistic reuse because components are sought from general reuse libraries in an effort to match them to a set of given requirements. Ad-hoc reuse has been very much the state of the practice in the industry.

The *by-technique* facet defines what approach is used to implement reuse. Compositional reuse is the use of existing software components as a building block for developing new software systems. Generative reuse is the reuse at the specifications level by means of applications level or code generators. It offers the highest pay-off potential, but at the same time it is a difficult technology to scale up to industrial productions.

The *by-intention* facet defines how reusable elements will be reused. Reuse components can be used as they are or with some modifications. These approaches are called black box and white-box reuse respectively.

The *by-product* facet defines what is the software work product that is reused in developing new software systems. The most common one include source code, designs, specifications, objects, text and architectures.

In this study, the reusability by-technique is composed of composition and generative we are trying to adopt. In the next section, a brief explanation about reuse by composition will be given.

2.3.2 Reuse By Composition

Reuse by composition means the developer selects components, and uses the selected components to build or compose a new application [32]. In general, a reusable component can be defined as a unit of design (at any level), for which a structure is defined, a name identifying the component is associated, and for which design guidelines, in the form of design documentation, are provided in order to support the reuse of the component and to illustrate the context where it can be reused [35].

From this general definition, reuse by component or composition can be defined from two different point of views [35]:

i. Design for reuse : approaches in which components are developed in such a way that they can be reused in different contexts

ii. Design by reuse : approaches in which software is designed, assembled and personalized.

The reuse by composition model provides more general and less domain specific sort of reuse [32]. This model sets no limitations to what should be reused or in what phase reuse should be practiced. In this way, the developer carries out the task of selecting components and makes them work together. Since in this model, the developer must have direct access to the reusable components, there are the criteria that the components can fulfill. Each component must:

i. Be well defined, perform a certain independent task
ii. Have clear and simple interface to the environment
iii. Be well documented

These requirements are important in order for the developer to understand, evaluate and incorporate the component into the new product and at the same time reduce the development cost.

2.3.3 Applying Reusability in Developing ITS

Since early 90's, ITS community realizes the importance to make ITS usable again and again. The main reason why we need to adopt reusability when developing ITS is because ITS software development is rather large as it requires huge computing resources, expert manpower and time for developing it [24].

El-Sheikh and Sticklen state the serious problems exist in the current technology in developing ITS in each application is that, it is developed independently and the tutoring expertise is hard coded into individual application [3]. There are a little reuse of tutoring component either student model or expert model or tutoring model or interface model between existing application due to lack of standard language for building each component [3].

In order to overcome this problem, researcher have tried to adopt reusability concept in developing ITS and evaluate the effectiveness of this new approach. They agreed by adopting reusability elements, time and cost for development can be largely reduced if existing knowledge can be reused to make another set of ITS.

Arruarte proposes three different ways of considering the reusability in developing, ITS which are; (i) direct reusability of knowledge means systems that use knowledge already existed; (ii) tools for promoting reusability means tools that provide facilities in order to reuse knowledge and (iii) tools for developing and generating application code [24].

By using these different ways to promote reusability elements in development, they have investigated two levels of reusability. Firstly, a case based instruction planner as a reusable component that can be integrated into several different ITS and secondly, case based instructional planner construct plans by using lesson plans previously applied in other learning sessions.

Some other researchers proposes a new framework or architecture to develop ITS to be reusable. For example, Tam proposes a reusable architecture named Welding Intelligent Training System (WITS) for developing VR based ITS that provides hands-on experience in appropriate virtual selling step by step, interactive training of personnel in alumino-thermal welding [53].

WITS-KBES is designed with the capability to reuse the existing knowledge that is stored by KBES. The knowledge can be modified or updated by a welding expertise or based on updates of the training session. Truly, WITS offers reusability since the data file can even be replaced by another training course information that has been formalized along using the same scheme [53].

El-Sheikh and Sticklen have introduced their architecture capable of generating intelligent tutoring system for different domains by interfacing with existing generic task-based expert system and reusing the other tutoring components by using an existing expert system [54]. The aim of this work is to reuse ITS from the leverage of the generic task (GT) development methodology. GT is their past work on expert system development methodology [3].

Their approach leverages the knowledge representation and reasoning capabilities of GT knowledge-based system, and reuses them into tutoring. The architecture employs a run able deep model of domain expertise, facilities fine grained student diagnosis that can offer an easy method for generating ITSs from expert system, and will allow the core ITS shell components to be reused with different knowledge bases among other features.

The architecture developed focuses on hierarchical classification, a knowledge representation and inference technique for selecting among a number of hierarchically organized options. Knowledge is organized in the form of hierarchy of pre-specified categories. The architecture of this work consists of three main components, which are GT expert system, an extended domain knowledge component and ITS shell. Figure 2.3 shows the architecture for generating ITSs from GT-Based expert system [54].

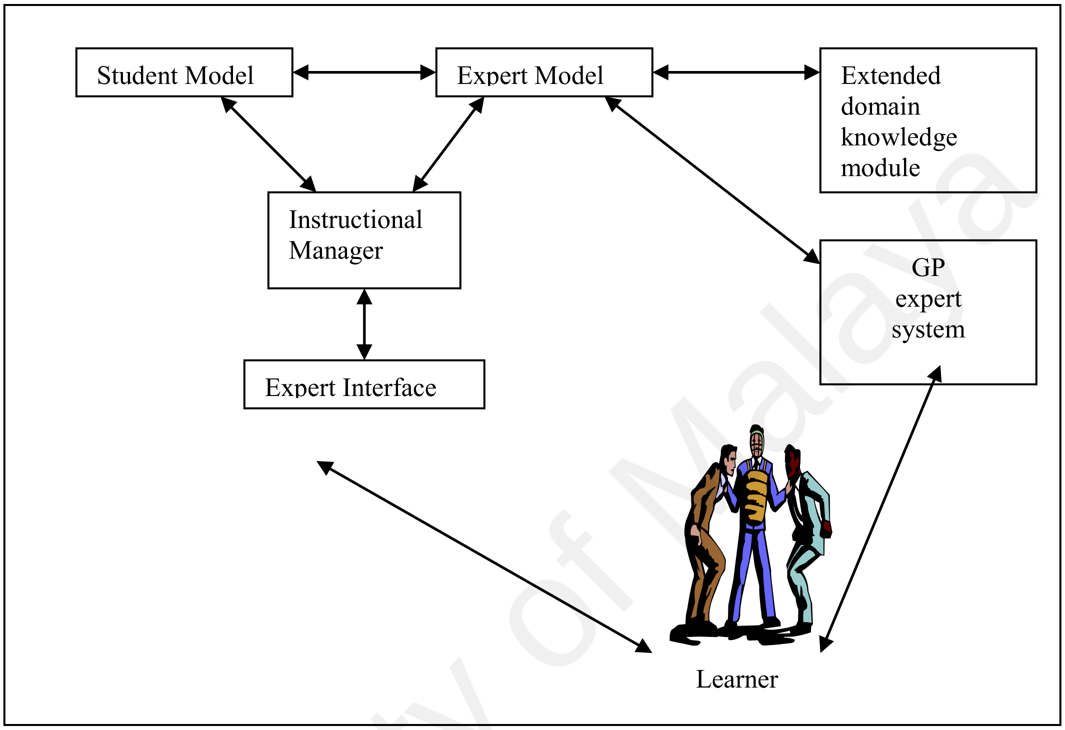


Figure 2.3 :
Architecture For Generating ITSs From
GT-Based Expert System

2.3.4 Building Expert Module as a Reusable Component

Expert module or knowledge module is the heart of ITS component [1]. Expert module embeds sufficient knowledge of a particular domain or topic to provide the ideas of answers to questions and allow the ITS to demonstrate or model a correct way of solving the problem, in fact, capable to generate different answer paths or goal structures.

Ideally, expert module can be derived from a tool that applies reusable approach or technology to generate it in a different domain. That means, we are providing a tool where the expert of domain knowledge can create his or her own expert module in different domain.

Murray believes that by using authoring system in the curriculum sequencing, tutoring strategies, and multiple knowledge type categories, it allows author to present simple facts, relationships and procedures in order to develop his or her knowledge base as a separate component to integrate with another ITS component[56].

El-Sheik and Sticklen in their work, show how to generate a different expert module before integrating it with another component to come out with other tutoring system [54]. They used Task Specific Approach to generate ITS in this work. Figure 2.4 shows the interaction between expert module with ITS and learner independently.

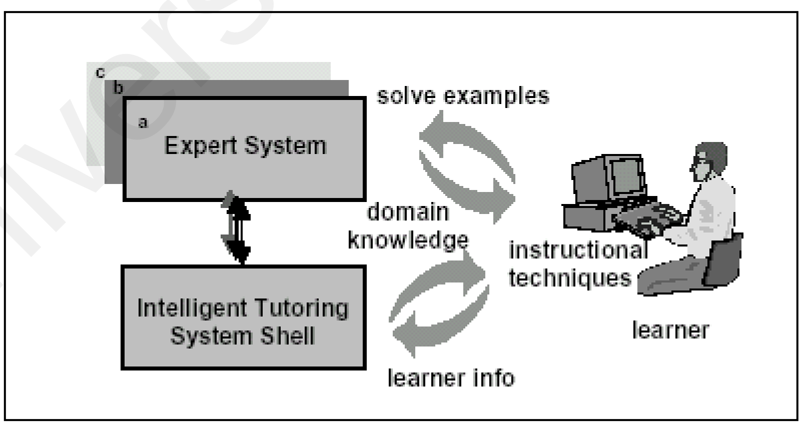


Figure 2.4:
Expert Module in ITS Environment

Many researchers have proposed to make another module called expert system where the expert system stores domain knowledge to generate an expert module. Once the expert module is completely generated, it will be integrated in intelligent tutoring system shell to be used by the learner.

On the other hand, some researchers' work toward developing a task-specific authoring tools that is capable to be reused, such as IDLE-Tool. IDLE-Tool stands for Investigate and Decide Learning Environment; capable for supporting the design and implementation of educational software for investigation and task decision [70]. Beside IDLE-Tool, researchers have made a study on another task-specific authoring development, TRAINER which is a shell for developing training systems for tasks such as medical diagnosis [70].

Why task-specific authoring? El-Sheikh believes task-specific approach offer considerable flexibility, while maintaining rich semantics to build knowledgeable tutors. Due to these advantages, task-specific approach becomes the best way to develop reusable software as a tool for developing ITS for him.

Experts who used this framework or architecture for developing their ITS will be interacting with the expert system from GT-knowledge based that adds the knowledge before deriving other components in ITS. In this stage, the expert module will be extracting three types of knowledge, which are, firstly, the decision-making knowledge, secondly, knowledge of the elements in the domain database and thirdly, knowledge of the problem solving strategy and the control behavior [3].

For this study, prototype tools that will allow experts to generate their own expert module will be developed. In this study, Qualitative Process Technique has been chosen to represent the knowledge with an assumption that the expert who will use this tool knows how to use QPT in representing the knowledge.

The question now is, why QPT? QPT can be defined as a process of central to human understanding of the physical world all-physical changes are caused by processes [48]. QPT is capable to present and to reason out behaviors of physical system without the kind of precise quantitative information needed by an expert. Qualitative reasoning captures people's ability to reason out and interact appropriately with commonplace physical situations.

In this study, QPT proposed by Forbus is adopted. Forbus suggests the representing knowledge in QPT should consist of five stages which are; (i) individuals view, (ii) preconditions, (iii) quantity conditions, (iv) relations between individuals and (v) influences that affected individual or the processes [20]. The explanation for each stage will be given in the next chapter.

In order to produce an expert module using expert module tool, an expert should have background knowledge in qualitative process techniques. He or she should have the knowledge and experience in how to break down the knowledge into stages that have to be declared. An expert will impart his knowledge in graphical form, represented by node in the forms of circle and arcs. The node represents the input or outputs in the form of object from the processes meanwhile arcs represent the relationships between objects that give the influence to the process.

From this graphical view, narrative text will be generated to produce a simple note about that processes. This notes will be used as a component in developing ITS and available to be used for different ITS as long as the ITS is in similar subject.

2.4 Overview of Knowledge Engineering

Knowledge representation (KR) is a part of an expert system, which is one of the components in Artificial Intelligent (AI). A knowledge representation is about finding an appropriate and an effective way to present the knowledge to an audience. So, how do we represent knowledge to the people so that they could that it understand is a challenging task but interesting to be explored by researchers.

Howe has defined AI as a part of computer science concerning concepts and methods on two major disciplines, first the symbolic inference by computer and, second symbolic knowledge representation for the use in making inferences [56]. In the same article, Howe defines knowledge as an object, concept and relationships that is assumed or considered to exist in some specific area or domain.

Knowledge represented in an expert system is concerned about with finding ways in which large bodies of useful information can be formally described for the purposes of symbolic computation. Formally described means; rendered in some unambiguous language or notation that has a well-defined syntax governing the form of such expression to a well defined semantic which reveals the meaning of such expressions by virtue of their form [59].

Basically, there are four main sources of knowledge. Firstly, documented sources which is found through books or manuals or etc. Secondly, knowledge retrieved from undocumented sources such as from people's minds or form machines. Thirdly, knowledge acquisition from database and the last source is knowledge acquisition via an the internet.

This section will review briefly the topic of knowledge representation including types of knowledge, how to map knowledge into QPT and related research on this area.

2.4.1 Type of Knowledge

Knowledge is a collection of facts from some domain or area. Some articles have defined knowledge as collections of specialized facts, procedures and judgment rules [59]. From this definition, knowledge can be concluded as a piece of information for a certain domain that includes facts or procedures or judgment rules. Researchers have divided knowledge into several types or categories based on what the knowledge is about and how deep the knowledge is covered.

The knowledge can be divided into two levels, which are shallow knowledge and deep knowledge [59]. Jackson has defined shallow knowledge as a knowledge, which consists of more ad-hoc linking of stimulus and responses to their environment. Meaning, shallow knowledge is more generalized knowledge consists of more information about broad domains. We can call it as surface knowledge, covers many topics without going into more details of that domain.

Deep knowledge can be defined as a special knowledge representing methods such as semantic networks or frames to allow the implementation of deeper level of reasoning such as abstraction and analogy activities. These activities are considered as important activities in developing expert modules or systems. Sometimes, deep knowledge is known as background knowledge in the expert system literature [60]. Deep knowledge using objects and processes in the domain of expertise in this level shows relationship among objects, which is important to be considered.

From these levels of knowledge, basically knowledge can be divided into three major categories, which are procedural knowledge, declaration knowledge and meta knowledge.

Procedural knowledge consists of the rules and skills use to manipulate and transform declarative knowledge. It includes the rules of mathematical and logical operations; the rules of grammar, inference, and judgment; strategies for forming percepts, encoding and retrieving memories; and motor skills. Procedural knowledge may be represented as productions. Procedural knowledge suitable to represent skill, strategy and process.

Procedural knowledge representations differ from declarative knowledge, which stores knowledge in a more flexible but harder to use format. Knowledge is represented in sequence with a number of steps. One advantage of procedural representation is the possible speed usage in a performance system. Productions are common means of representing procedural knowledge.

Meanwhile, declarative knowledge is in the form of statements about truths whereas there are the know-what rules. Declarative knowledge tells us why things work and the way they do, or that an object or a thing has a particular name or location. It includes information about concepts and elements in a domain and the relationships between them. Declarative knowledge is more flexible than procedural knowledge. It allows us to make predictions about the future and to explain why things happen.

Meta knowledge is knowledge about knowledge. Specifically, it is knowledge about what we know and what we don't know [69]. This applies to an individual and an organizational level. Menzies give a definition for meta knowledge as referring to knowledge that has been acquired and stored in prior system development activities and that is applied to the current software design project to improve the quality of the end product and to reduce its cost [41].

2.4.2 QPT as a Tool To Present The Knowledge

One of the tools or techniques that can be used for presenting knowledge in developing an expert module or system is a Qualitative Process Technique (QPT). QPT is the process central to human understanding of the physical world. All-physical changes are caused by processes [48]. An example of a process in which QPT are needed in representing knowledge is boiling water, or flow of fluid.

QPT is introduced by Forbus in 1984 based on qualitative reasoning concept [20]. The Qualitative Process Theory defines a simple notion of physical process that appears useful as a language in which to write dynamical theory [20]. Meanwhile, qualitative reasoning focused on guiding the application of numerical techniques by helping with the quantitative model selection or parameter setting [63].

Recently, qualitative reasoning has attracted much interest from the artificial intelligent research community. Qualitative reasoning is aimed at developing representation and reasoning techniques that will enable a program to reason out the behavior of physical systems [66]. As such, qualitative reasoning is becoming one of the important issues to consider as a domain in AI research today to produce better quality expert system.

Basically, there are four basic techniques in the qualitative-reasoning technique [66], they are:

i. Qualitative arithmetic

It is arithmetic of sign or, more generally of interval. In qualitative arithmetic, the variables take the values of positive (+), negative (-), or zero (0). The definition of qualitative arithmetic operators is shown figure 2.5.

		x		
		+	0	-
y	+	+	+	?
	0	+	0	-
	-	?	-	-

x[+]y

			X	
		+	0	-
	+	+	0	-
y	0	0	0	0
	-	-	0	+

x[*]y

Figure 2.5:
Definition of Qualitative Arithmetic [66]

ii. Qualitative simulation

Qualitative values give only minimal information about the state of the system being modeled [66]. It is becoming more useful when some of the variables are datives, because their qualitative values indicate whether or not a variable is increasing. Through simulation, values that are indicated could be shown to be more useful and helpful.

iii. Knowledge representation

It allows us to make inferences about the given physical system or situation. Simulation is one of the effective ways to represent knowledge. There are two main concerns that dictate the design of knowledge-representation formalisms in qualitative reasoning:

a. Generality – representation should be general enough to represent a wide variety of physical phenomena, both in different theories about the world.

b. Modularity - representation should be modular. All modules should be small enough and independent enough that they can be reused in different combinations to compose complete models of different physical situations.

iii Model formulation and abstraction

Appropriateness of a model in this case depends on factors such as whether the model includes relevant information or is at the right level of abstraction for answering a given analysis questions. How do we know whether or not a given design will be successful in projecting an image on the screen? To make it successful, the model should include the optical aspect of the device being designed.

So, based on these four basic qualitative reasoning techniques, QPT comes in five stages - individuals, set of precondition, set of quantity conditions, set of relations and set of influences in representing knowledge. Physical process means something that acts through time and change parameters of objects in a situation [20].

Processes of QPT begin by specifying the individuals involved in the process. Each individual could be an object. The object will be defined as individuals similar to variables in a programming language where we assign one variable as an alibi or representing another object. For example, we could define individuals as below:

```
Individuals:
src an object, has quantity(src, heat)
w a contained-liquid.
```

Figure 2.6
Declaration for Individual Stage

The next process is defining a set of preconditions for each individual. Preconditions explain what the prerequisite conditions must have at the individual level for the declared object to make sure the next process could be done successfully. Preconditions will also define the relationship between one individual to another individual.

The next QPT process involves defining a set of quantity conditions. Quantity conditions explain either statements of inequalities between quantities belonging to the individuals (including domain-dependent constants and functions of them) or statement about the status of processes and individual views.

The next process which follows is the quantity conditions, which is a process for defining a set of relations where this process imposes on the parameters of the individuals along with any new entities that are created. Relations will define attributes and the type of relation between objects. In this process, each object will declare its relations and how it relates to another object.

The last process will be a process to identify the influences of each object. There are two main influences, which are direct influence, and indirect influence. A set of influences imposes the process on the parameters of the individuals. Influences specify what can cause a quantity to change. The influences component of a process specifies the direct influences imposed by the process. Below is an example on how heat flow process can be presented using QPT adopted from Forbus' paper that discusses qualitative process [20].

Process heat-flow

Individuals:

src an object, Has-Quantity(src, heat)

dst an object, Has-Quantity(dst, heat)

path a Heat-Path, Heat-Connection(path, src, dst)

Preconditions:

Heat-Aligned(path)

Quantity Condition:

A[temperature(src)] > A[temperature(dst)]

Relations:

Let flow-rate be a quantity

A[flow-rate] > ZERO

Flow-rate $\propto_{Q+}$ (temperature (src)-temperature (dst))

Influences:

I-(heat(src), A[flow-rate])

I + (heat(dst), A[flow-rate])

Figure 2.7 :
Example of Presenting Process Using QPT

In this work, QPT approach is used in presenting knowledge about the process . An experts will declare his object, by constructing a circle at graphical view. After that, he will complete the information about each object such as their preconditions, quantity condition before establishing relationship between one object to another object. When establishing the relationship, he needs to identify either the relationship will affect other object directly or indirectly.

2.4.3 QPT Modeling in Education

Education is an activity of educating or instructing or teaching that will impart knowledge or skills to individual. In order to ensure the delivery of knowledge, many methods or styles of delivers are explored by educator or instructor in manual form. In computerize form; numerous approaches have been explored to develop effective education tool or software. Many researchers either from education field or computer field have studied a lot of method in modeling knowledge before it is delivered to the students.

Nowadays, QPT has become one of the modeling techniques in order to deliver knowledge. Forbus and Bredeweg agree that qualitative technology is valuable for education because two main reasons; firstly they agree that much of education is concerned with conceptual knowledge. Secondly, the qualitative technology can provide the necessary ground and framework for quantitative and traditional mathematical models [72].

Many of researchers have studied how to model the knowledge in QPT and translate it into computerize form, mainly in software or tool of application. From here, students will be able to learn by themselves with the guidance of application to explore the knowledge. For example, Salles et.all has discovered how to model Portuguese language as a second language for deaf student using qualitative reasoning [73]. In this study they have used three models in order to present the diagrams of knowledge. The models used are; the Growing Three model, the Drying Shirt and Cataguazes model [73].

Syed, Phang and Sharifuddin have introduced the application of qualitative process theory in chemistry topic [71]. They have developed the software that is capable to do the simulation of chemistry reaction. With this application, students can use a qualitative model alongside their classroom experiment to determine optimal mixture of substance. The qualitative model of this application will show the end result using color coding and explain why the result is obtained.

Salles et. all have extended the previous researchers work, which is to model the Daniell cell for chemical education using qualitative model [74]. Through this work, they had chosen the process centered approach because it allows for the explicit representation of objects, quantities, possible value, processes and quality. The models were implemented in GARP for presenting the selected knowledge.

Currently, there are many researches going on to explore and enhance the capability of qualitative technology for education purpose. Through qualitative reasoning research, researchers have developed theories, representations and reasoning techniques that enable user to address the issues. Gradually, they were achieved successful result in their researches to address QPT as one of the educational methodologies.

2.4.4 Develop Expert Module Using QPT

Several researches have been done by researchers in order to adopt QPT technique to develop expert module for ITS. For example, Bredeweg developed Visi-Garp, which is a tool capable to support investigation of the simulation model and results [25,67]. In developing Visi-Garp, Bredeweg had used qualitative simulation based on qualitative reasoning theory and technique. Basically, by using this approach, Bredeweg had represented the content for simulation by encoding the study explicitly in term of structure, parameters and causal [67]

In order to visualize and present content as above, Bredeweg had used several symbols such as circular, rectangular and oval to represent the entities. Meanwhile, lines, arrows inclusion, ordering and indentation represent the relationships between entities [67]. How did Bredeweg build the model using QPT by using GARP? Actually, Bredeweg had used MOBUM, a separate tool in order to develop or design the knowledge into QPT form [29].

MOBUM is a domain independent model-building environment aimed at supporting learners in building qualitative simulation. MOBUM has four builders in order to allow the learner or expert to generate their own simulation. Experts firstly need to draw a physical model of knowledge at a workspace using objects and arcs. At the same time, the hierarchy of relationship between these objects will be determined as well. From the physical model, experts will create an ordered set of quantities values that may be presented as well as the specific features and instances. Then, the content that need by expert will be representing using graphical view at the workspace.

After finished creating the view by declaring objects and establishing arcs between objects, this work, MOBUM will be saved before integrating it with GARP to become a complete ITS. The weakness of this final output is there is no explanation about the graphic that has been drawn. This situation is difficult for any users that might use GARP especially those who are not familiar with QPT symbols and techniques.

Hence, in this study a prototype tool to generate graphical view as well as MOBUM will be developed and tried by using narrative text that will explain what is represented in graphical view.

2.5 Discussion and Conclusion

This chapter formally has discussed about three main ideas in this work that are Intelligent Tutoring System, reusability concept and knowledge representation in developing a reusable expert module as a component in ITS. Developing ITS recently comes with two major problems which are very costly and time-consuming for developing and it also involving too many experts in order to build just one ITS for one subject.

On the other hand, miscommunication between the developers, who is responsible to develop this expert module and the expert; who provides the content of the module has always become a major problem in the development phases. One of the solutions for this problem is to develop the tool that allows the expert himself to produce his own expert module.

In order to overcome these problems, El-Sheikh introduced task specific authoring as a general guideline to develop expert module for different domain [54]. Meanwhile Tam had introduced framework concept to generate different expert domain named Welding

Intelligent Training System (WITS) to developed VR based ITS [53]. From these two different studies, researchers tried to develop a generic framework, which enable experts to build their own expert module by providing related information.

By using QPT as a tool to present content by an expert, Bredeweg and Vania [29] had developed MOBUM to visualize content into qualitative simulation. The final result from this work is graphical views consist of arc, circle and text presenting the knowledge that have been coded by the expert.

As a derivation from this idea, this study intent to develop the tool prototype that is capable of generating different expert module for different subject using QPT approach by the expert himself. This prototype tool will try to adopt reusable concept in order to allow generating different expert module for different subject. The difference between this tool with MOBUM is that, this prototype tool will generate the narrative text to give a simple explanation about the graphical view that is available at workspace.

Perhaps this idea can give an alternative solution in order to avoid expensive cost and save time as well as miscommunication between a developer and an expert to develop an expert module for ITS. The next chapter will discuss on how the prototype tool is developed and how the QPT framework is mapped for different subject as well as the results from this work.

Chapter Three

An Architecture of Prototype

3.1 Introduction

In this study, a prototype tool, which allows users to build an expert module component that will integrate with other components to produce a complete ITS is developed. The prototype tool is reusable in the sense that it can be used to produce several different expert modules for different ITSs. Qualitative Process Theory was used as a general template to present knowledge in the development of the expert module.

Qualitative Process Theory was used as a general template for knowledge representation for the expert module as it has proven its ability to model several types of domain application event to simulate virtual reality using 3D approach [62,67].

Can a tool to generate an expert module with the following criteria be developed for different subjects? In must : (i)Capable to present knowledge using QPT approach ; (ii)apply the reusability concept in constructing tool and generate expert module; (iii)be able to present knowledge in graphical form and generate the relevant meaning according to the graphical form developed.

In the context of this study, reusability refers to the tool's capability to produce different expert module rather than reusability of knowledge content. The difference between this tool and the conventional tools is that the former produces different types of expert modules and content knowledge for different domain using the same tool, while the latter requires a different tool for each particular expert module.

Knowledge experts in a particular domain are required to have an understanding of QPT in term of its symbols, concept, and relationships between symbols and how they work. Knowledge representation is basic and primitive to allow versatility in presenting wider range of domain application. However, a more user friendly function that allows natural language expressing an expert knowledge could possibly be developed in the future.

3.2 Overview of the Prototype

This section will give a general idea or an overview on how the prototype is working. Basically, the main concern in developing expert module using this prototype is the way we convert information from raw knowledge into QPT syntax. But, as we mentioned before, we are not discussing on how we are going to convert this raw knowledge into QPT syntax because our intention is to build an expert module using this approach. So, we assume that experts know how to convert it into QPT syntax.

The main input to generate this expert module is QPT syntax that consists of object name, pre-condition, quantity condition, attributes and relationship between objects. Figure 3.1 illustrates the process on how the expert module could be developed by expert using this prototype.

In the stage, an expert needs to define all the objects that are available at individual stage in QPT syntax. Each object will be represented by one circle which is labeled. Then, he or she needs to define, for each object, the pre-condition, the quantity condition and the defining attributes.

Before he or she completes generating a graphical view, he or she needs to establish the relationship between objects. When all the objects have their own relationships with another object, he or she will generate the narrative text that is able to be saved and used as an expert module in ITS.

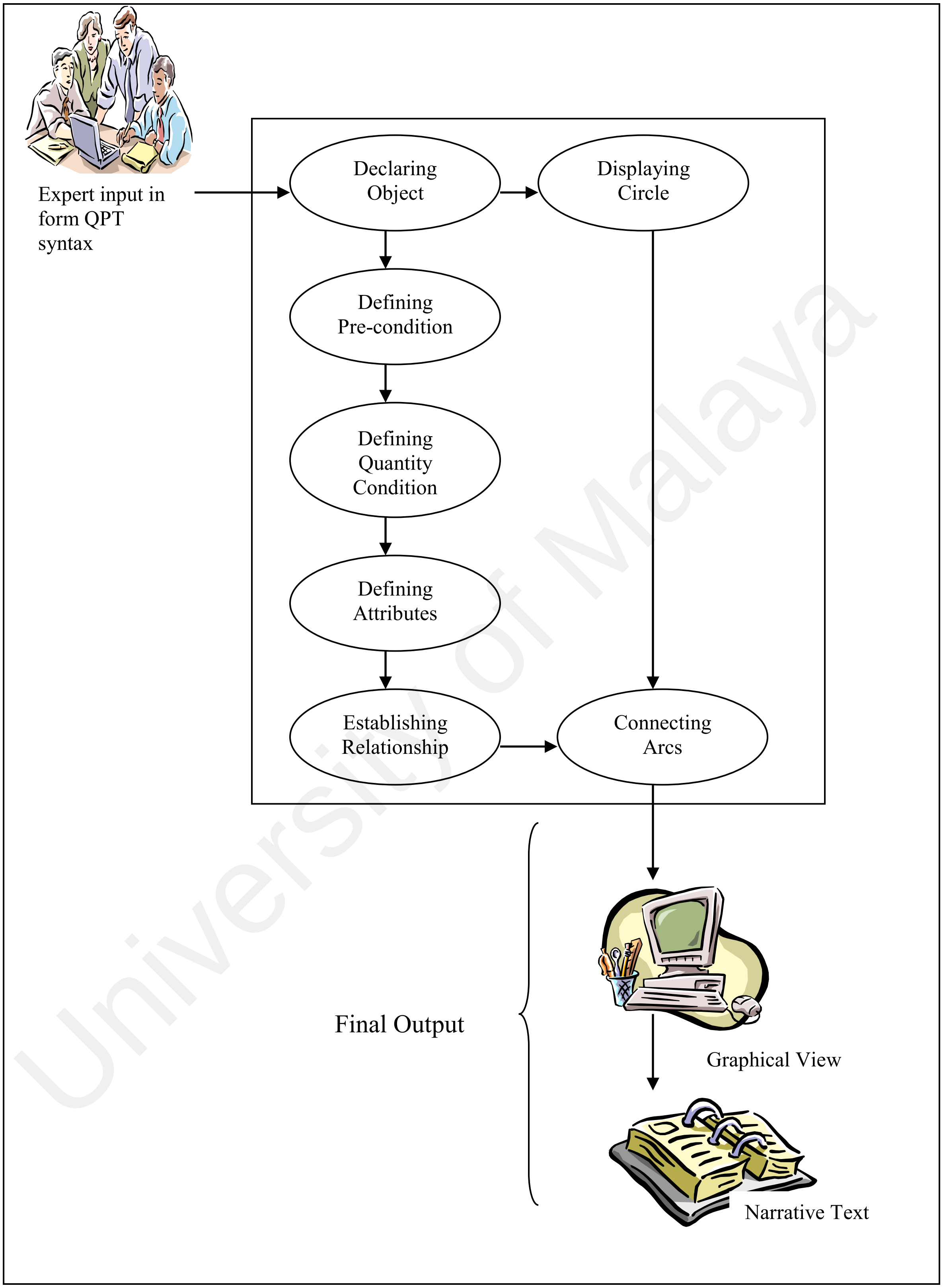


Figure 3.1 :
Overview of Prototype

3.3 The General Architecture of Prototype

In this section, we discuss the architecture of the tool prototype and how to map or represent the knowledge using the QPT approach. This can be achieved through graphical manipulation using arcs and nodes to represent symbolical meaning. The graphical representation is the process to generate a narrative text about the knowledge being modeled.

Generally, a prototype tool can be used to produce an expert module. This prototype contains two basic features, namely the graphical view on how the knowledge is to be presented and a narrative text that translates the graphical view into a language form.

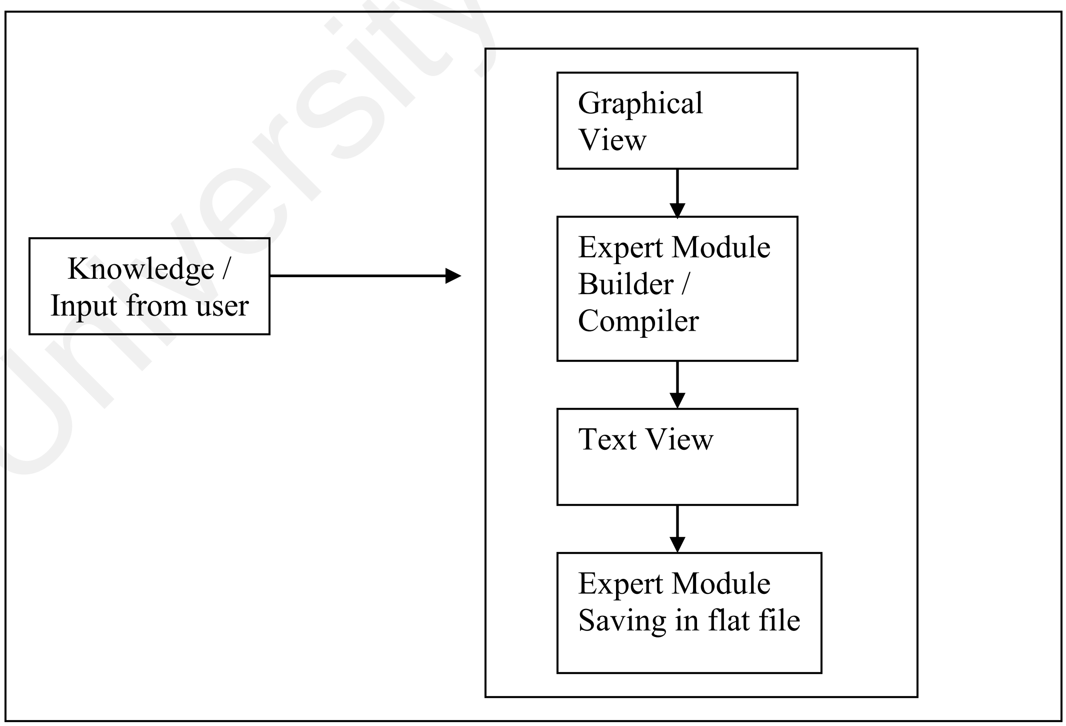


Figure 3.2 :
General Architecture on Developing Expert Module

Mainly, a builder for generating from user input to narrative text is the main part of this prototype. The engine that we develop to translate the knowledge from graphical view to narrative text will read input that is defined by experts and arrange into a format that can be read or understood by the common user rather that graphical view itself. Figure 3.2 presents the general architecture of this tool prototype which comprises namely component graphical view, expert module compiler and text view in narrative text format.

Why do we need to translate graphical view into narrative text? Basically end users have difficulties in understanding knowledge that is represented by graphical view. In graphical view, nodes represent objects and arcs represent the relationship of one object with another object. Arcs and nodes are the simplest symbols to represent and express the concept of QPT rather than using other symbols [67].

A graphical view that represents certain knowledge domain will be converted into narrative text. Narrative text form will help end users to understand better the narrative knowledge that is presented. Thus, the final output of the tool prototype is the graphical view that includes arcs and nodes in the form of graph and narrative text that describes the graph.

3.3.1 Steps in Generating Expert Module

In order to build this expert module by using QPT approach in terms of our definition, experts needs to go through several steps. Starting with declaring object, precondition, quantity condition, attributes and finally by establishing relationship between two declared objects. Figure 3.3 shows a flow of generating view from plain knowledge to a narrative text. Each of these steps will be elaborated in more detail in the next chapter.

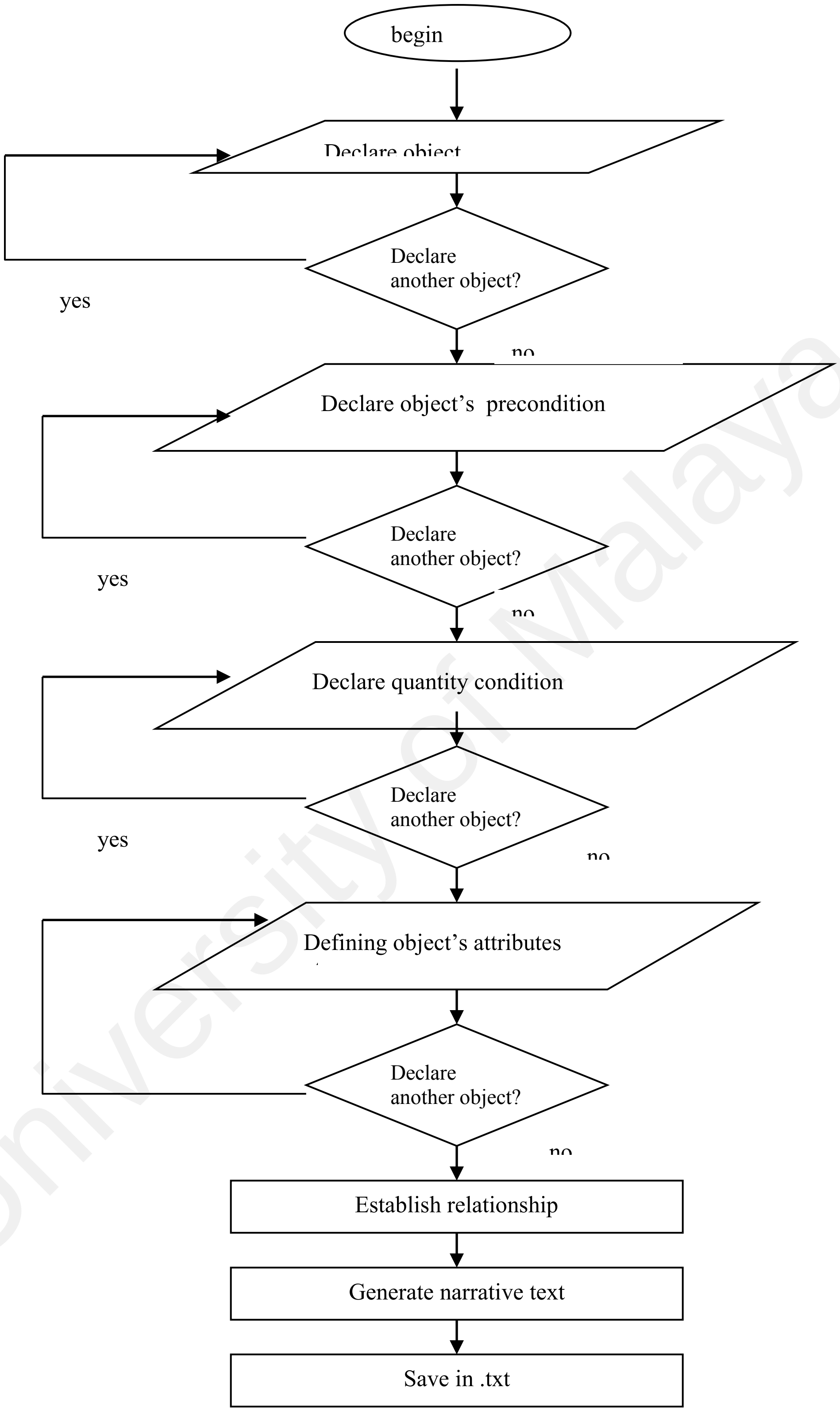


Figure 3.3:
Flow of Producing An Expert Module

The process of building an expert module begins when experts formalize or model the knowledge into QPT syntax in the text based. After that, experts will translate this syntax into a graphical form using arcs and nodes in the form of a graph that will be converted into the narrative text and will be saved in the text format.

The expert will declare or create his objects in graphical view and will represent it in a circle form. From this object, the expert will declare or assign attributes, pre-conditions and quantity conditions that must be fulfilled for each object as a requirement to make sure that the knowledge is fully transferred into a graphical form.

After declaring this entire requirement for each object, the expert needs to establish a relationship between objects. Each object that has been declared should have at least one relationship with another object. If not, this object will not be detected in the narrative text form. So, basically the narrative text will only be generated when there are relationships between objects and their conditions.

The expert will then arrange the declared objects on a workspace. In this workspace, the expert will group the objects according to their levels. For example, if an object is declared in the individual view, this object should appear at the bottom of the workspace. How each object is categorized into the three groups will be discussed in the next subtopic.

Information about each object and its relationships with other objects will be stored in files according to the types of information. The expert module compiler or builder will read all the information and organize it before translating into narrative text form. This form will be saved in another file, which will become a component that can be integrated with other components to develop a complete ITS system. For this study, the expert system is built in narrative text form only. However, for future research, generation of this knowledge in other forms such as images or simulations should also be considered.

Eventually, the expert module which is produced can become one of the components that can integrate with other components in developing ITS. For example, if we are developing an equilibrium expert module, this module might be used again and again as a component in developing chemical Intelligent Tutoring System for different topics.

3.4 Representing Knowledge in QPT Approach For Generating Text

In QPT, knowledge is represented in a basic form called individual process; It is comprises five parts that are the individuals or objects, a set of pre-conditions, a set of quantity conditions, a set of relations and a set of influences, which are divided into three categories of views; individual view (IV), view instance (VI) and process view (PI).

An object is an instance that can be created and destroyed, and its properties can change dramatically. Pre-condition will define the situation or state that must be true for the objects. The state or condition about the object itself is dynamic and can change according to the environment. This change occurs in the quantity condition phases. Thus quantity condition is a statement about inequalities between quantities of the individual and statement about whether or not certain other individual views hold. Relations will represent the physical relation between two objects to indicate the statement should be true whenever the view is true.

Individual view (IV) will illustrate the changing states of the object. Properties of the objects can be viewed in individual view. How does this work? To understand it, let's consider the water and a cup. We can pour the water into the cup and then heat it. Some of the water will evaporate and become a steam or gas, and the quantity of fluid water will be reduced. Then if we put something that has a low temperature at top of the cup, the steam will be converted back into fluid and the quantity of the fluid will be increased again. To model these kinds of changes we use individual view.

Every collection of objects that is been defined called view instance (VI). In VI, the status of precondition and quantity condition will be set as active, or inactive. Active status indicates a condition whereby the relation between individuals is specified.

On the other hand, process instance (PI) will represent objects that satisfy the individuals' specifications for a particular type of process. When PI is active, it represents the process acting between individuals' views (VI) whenever both the preconditions and the quantity conditions are true.

Let's consider boiling water again as an example to get better understanding of these concepts. Water, in the form of a liquid is a combination of two atoms of hydrogen and one atom of oxygen. In QPT approach, the water to be formed as a liquid is declared when the two objects, the hydrogen and oxygen exist in the individual view. For each object that has been declared, the pre-condition and quantity condition must also be declared to identify his or her attributes. In this case, pre-condition for boiling water process could be the object in a fluid form, and the temperature of the heat should be higher than the temperature of fluid. All of these processes will occur at individual view.

View Instant (VI) will define relationship between these fluids with other object such as the heat and will also differentiate the quantity conditions between substances. Heat flow happens between two objects that have heat and are connected via some paths through which heat can flow. When the water is heated, the liquid will subsequently boil, and steam or gas will be produced. This gas is made of the same stuff as the liquid but in a different form. This process will happen at instance level. Figure 3.4 shows how the process of boiling water can be represented using QPT approach.

```
Process Boiling Water
Individuals :
        w a contained – liquid
        hf a process-instance,  process(hf) = heat-flow
                                  ∧ dst (hf) = w

Quantity Conditions:

        Status (hf, Active)
        ¬ A [temperature (w)] < A [t-boil (w)]

Relations:

        There is g ε piece-of-stuff
        gas (g)
        substance (g) = substance (w)
        temperature (w) = temperature (g)
        Let generation-rate be a quantity
        A [generation-rate] > ZERO
        Generation-rate ∝Q+  flow-rate (hf)

Influence:

        I-(heat (w), A [flow-rate(hf)] )
        ; The above counteracts the heat flow's influence
        I – (amount-of (w), A[generation - rate])
        I + (amount-of (g), A[generation - rate])
        I –(heat (w), A[generation - rate])
        I – (heat (g), A[generation - rate] )
```

Figure 3.4:
Process of Boiling Water in QPT Syntax [adopted from 20]

3.5 Conclusion

Developing an expert module as one of the components in ITS is not an easy task. An expert should first determine who the users would be, the domain to be presented, and other considerations. To make this expert module a reusable component is as difficult as developing the component itself. However, with the current technologies and approaches in developing and representing knowledge, building an expert module as reusable component could be done.

This chapter has discussed how to build an expert module as a reusable component and how the QPT approach makes the representation of knowledge using a similar tool possible. This will benefit individual developers and companies who intend to develop ITS commercially.

Chapter Four

Results and Discussions

4.1 Introduction

This chapter offers an insight of the practices in developing expert module from the prototype tool that has been discussed before. There is a range of different approaches in developing expert module for representing knowledge in ITS such as rule based, semantic network and etc.

In the next sub-topic we will in discuss detail of how do we develop expert module for physics and chemistry. We have chosen the process of precipitate formation of iron sulphide (FeS) for chemistry subject and the process of heat flow for physics subject to simulate the production different expert modules using this prototype tool with QPT approach.

4.2 Developing Expert Module for Chemistry

Chemistry is the science of matter; the branch of natural sciences dealing with the composition of substances and their properties and reactions. In other words, chemistry is the scientific study of the compositions, structures, and properties of substances and the changes they undergo. Developing chemistry's expert module means dealing with changes of substances in either their characteristics or functions.

For chemistry subject, this study has chosen to represent the process of precipitation of FeS. The process is about dissolving ferum and sulphide ions using water as the medium. The process begins when there is a substance of H_2S and both the H^+ and S^{2-} are not saturated [71].

Figure 4.1 shows how the individual view is set-up by defining ferum as an object. Similarly, we can define sulphure as an object of the same individual view. In this step, expert needs to specify object's name, object's ID and view level, and where this object will be located. Object name is the full name of the object while, object ID is an abbreviation name used in graphical view. After the object is created, a circle representing this object will be created as shown in figure 4.2 .

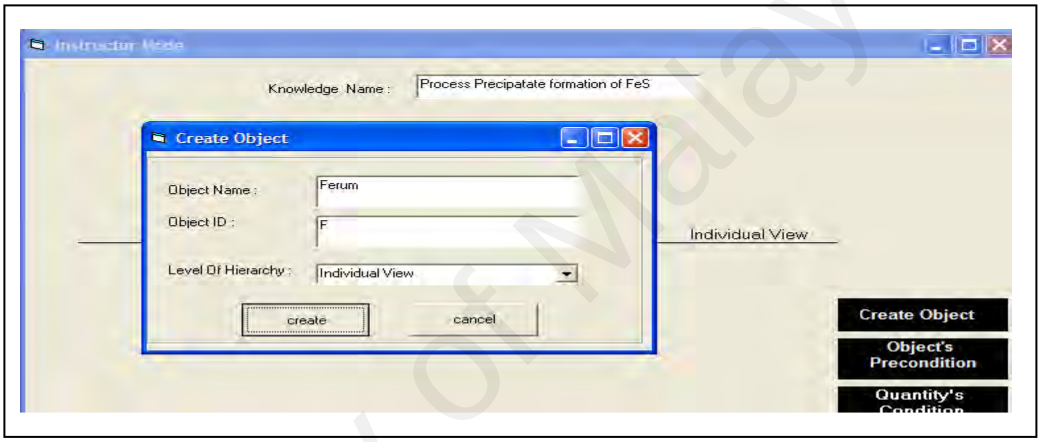


Figure 4.1 :
Window For Declaring Object

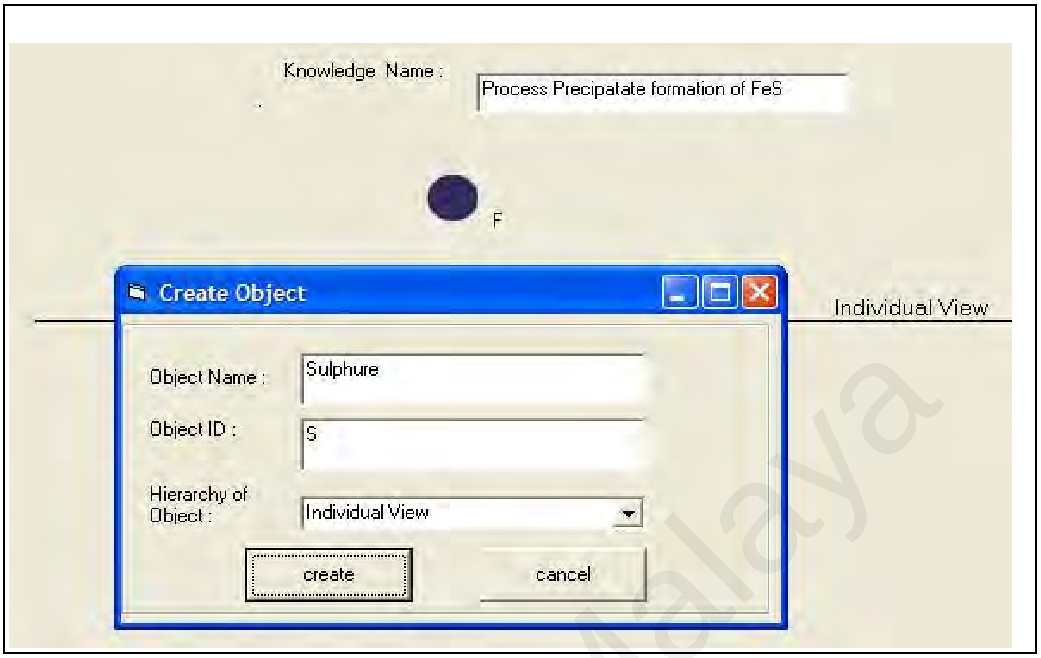


Figure 4.2:
Object Representing Ferum as a Graphical View

After declaring each object that is involved in this reaction, pre-condition for the object should be defined. Precondition is a statement about the object's individual state and their relationships other than quantity condition [20]. The pre-conditions and quantity conditions state all of the possible reactions that could occur with certain conditions. If a process does not fulfill the precondition, reaction between objects will not occur.

But, sometimes there is no need to declare precondition for each object. The precondition of an object need to be declared only when in that process it has certain conditions or some rules must be fulfilled before the reaction can occur. Sometime, precondition can be the main rules to ensure reactions can occur successfully.

When an expert has to declare pre-conditions for certain object, he or she needs to choose precondition button and the window as shown in figure 4.3 will appear. The, expert needs to select which object he or she wants to define its pre-condition. Next, the user will specify how many preconditions that object needs to have. Each precondition needs to be declared separately. A list of preconditions for the object that has been declared will appear at the precondition list box to ensure that the declaration is not repeated.

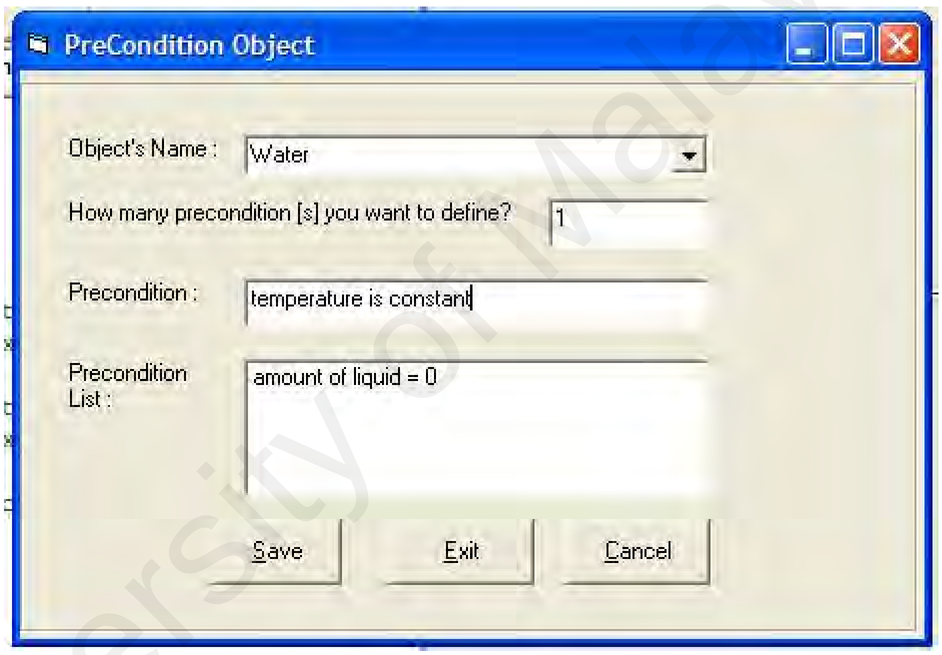


Figure 4.3 :
Pre-condition Window

Subsequently, quantity condition will be defined for each of the object. Quantity condition is a statement of inequalities between quantities of the objects or statements about the status of processes and individual view [20]. In certain cases, there is no need to declare quantity condition for the object that has been declared before. That means, not all objects that have been declared will have their quantity conditions to declare for.

Figure 4.4 :
Window For Declaring Quantity Condition For Objects

In declaring quantity condition stage, the user needs to declare each condition related to the object that has been declared before at each level of view. In the combo box list of objects as shown in figure 4.4, all objects that have been declared before will appear. For example, we have declared four objects at the declaring object stage for precipitation of FeS process, and all of these objects will appear at that combo box list.

The quantity conditions need to be declared separately for each object. The expert needs to select which object(s) have a quantity to declare. For example, if an expert chooses to declare quantity conditions for ferum first, the user needs to choose ferum from the list box before stating how many quantity conditions for ferum need to be declared. The expert is allowed to insert each condition that he or she wants to declare in the condition(s) box before it reappears in another box.

Next, we need to establish the relationship between two objects among the entire objects that we have declared before. According to Forbus, relationship means the process imposed by the process on the parameters of the individuals [20]. Relationship will represent what reactions occur between objects.

Types of influence or relation between objects occur directly or indirectly. Influence specifies what will happen if the quantity conditions change and how they affect the relationship. In figure 4.5, we specify how each object will affect another object either directly or indirectly in the influence menu.

The types of relationship will describe how object one reacts with object two. For example, one of the relationships for the precipitation process of FeS is K $\propto_{Q+}$ concentration-of (F) [71]. When declaring relationship for this relation, $\propto_Q$ will represent the type of relationship between object K and object F. Figure 4.5 shows the window for declaring or establishing relationship between objects in QPT.

Figure 4.5:
Window For Establishing Relationship

Defining object's attributes means users are declaring characteristics owned by the object. In many cases attributes for objects are not really needed as a consideration to make a process in the QPT approach. But, in some other cases, we need to define attributes for objects because sometimes, they can affect the reaction between objects.

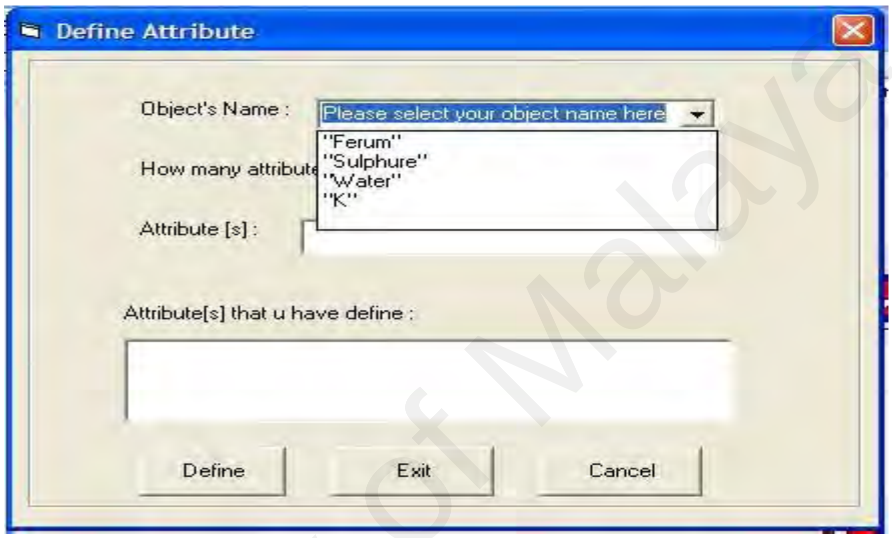


Figure 4.6:
Window For Defining Object's Attribute

In declaring attributes, user should follow similar steps as when declaring quantity condition. It starts with the expert chooses which of the objects he or she needs to declare. Then the user specifies how many attributes he or she needs to declare and write them separately in at the attribute column. List of attributes that have been defined will appear in a separate column.

After the declaration stage, the user will complete the graphical view that consists of object and relationship between objects, which can be viewed on a workspace. A graphical view representing how the process occurs is shown in figure 4.7. Each circle represents an object while the arcs represent relationship between objects to ensure that the reaction occurs.

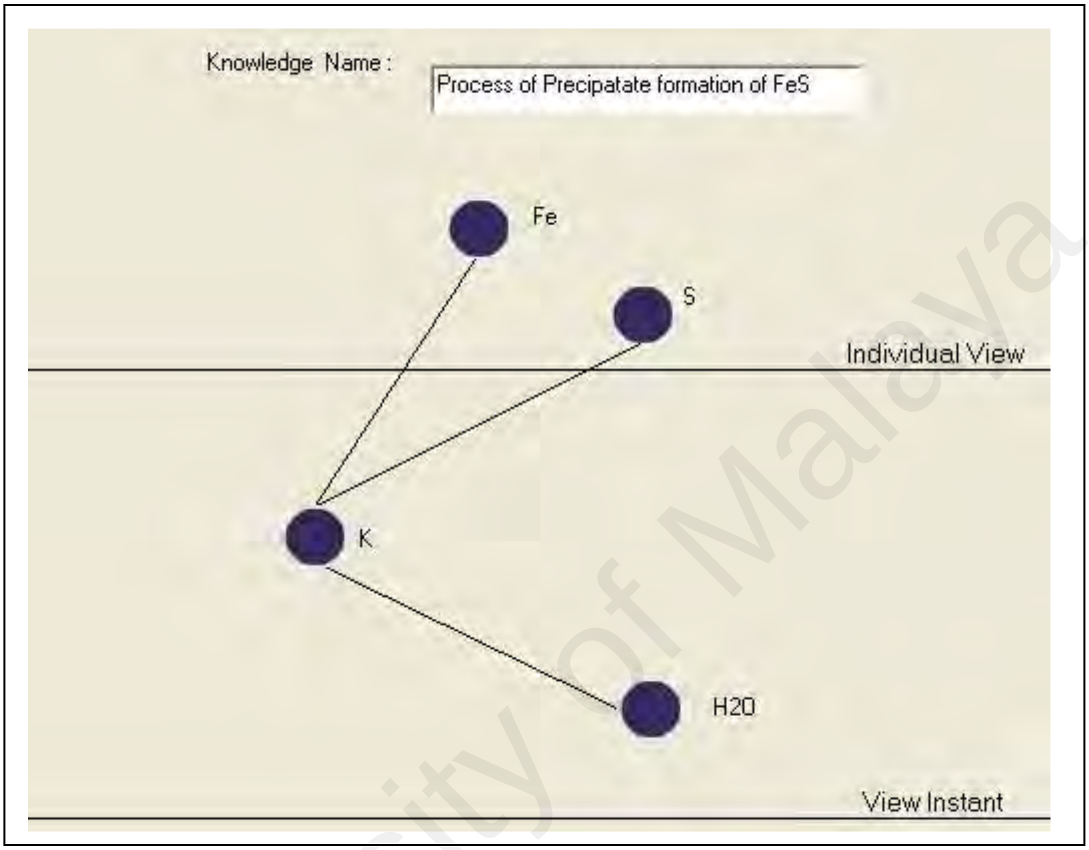
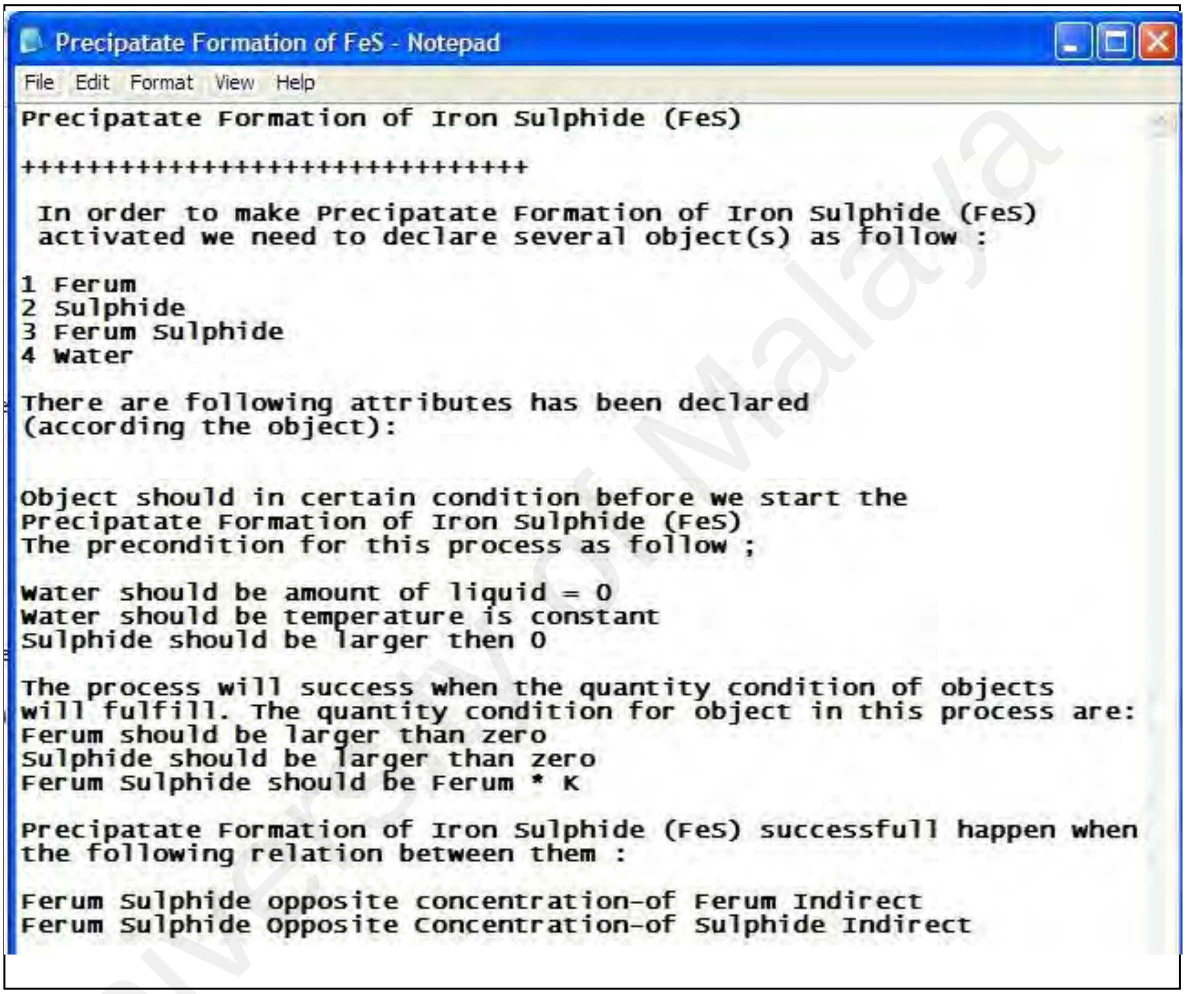


Figure 4.7:
Graphical View for Process of Precipitate of FeS

From the graphical view, the user can generate narrative text to explain the diagram. All of the information that was keyed during the declaration stage will be compiled and rearranged. This information will be saved in a text file. Figure 4.8 is an example of the text that is generated from a graphical view from figure 4.7.

This text file contains some information about the domain knowledge that we presented using the graphical view. When building a complete ITS, this text file could be used as one of the sources for building a complete expert module.



Precipatate Formation of FeS - Notepad

File Edit Format View Help

```
Precipatate Formation of Iron Sulphide (FeS)

+++++++++++++++++++++++++++++++

 In order to make Precipatate Formation of Iron Sulphide (FeS)
 activated we need to declare several object(s) as follow :

1 Ferum
2 Sulphide
3 Ferum Sulphide
4 Water

There are following attributes has been declared
(according the object):


Object should in certain condition before we start the
Precipatate Formation of Iron Sulphide (FeS)
The precondition for this process as follow ;

Water should be amount of liquid = 0
Water should be temperature is constant
Sulphide should be larger then 0

The process will success when the quantity condition of objects
will fulfill. The quantity condition for object in this process are:
Ferum should be larger than zero
Sulphide should be larger than zero
Ferum Sulphide should be Ferum * K

Precipatate Formation of Iron Sulphide (FeS) successfull happen when
the following relation between them :

Ferum Sulphide opposite concentration-of Ferum Indirect
Ferum Sulphide Opposite Concentration-of Sulphide Indirect
```

Figure 4.8:
Narrative Text Generated From Graphical View

4.3 Developing Expert Module for Physics

Most physics theories consists of characterization of processes through differential equations that describe how the parameters of objects change over time. In this section we will look at how we can develop an expert module in physics domain by using the same approach we have used to develop an expert module for chemistry.

In physics, we have chosen the heat flow process [20]. The heat flow process is a process of transferring heat flow from one object to another object. The objects should have different temperatures to ensure heat transfer can be done. Heat path is a heat connection between these two objects.

At the beginning of stage the an expert needs to specify how many objects is involved in this process and declares them separately at the declaring object window. The expert will be prompted to declare objects in the window as shown in figure 4.9. He or she needs to specify object ID and level of hierarchy where the object is located either in the individual view or the process view or the instant view. A circle representing the declared object will appear at the workspace.

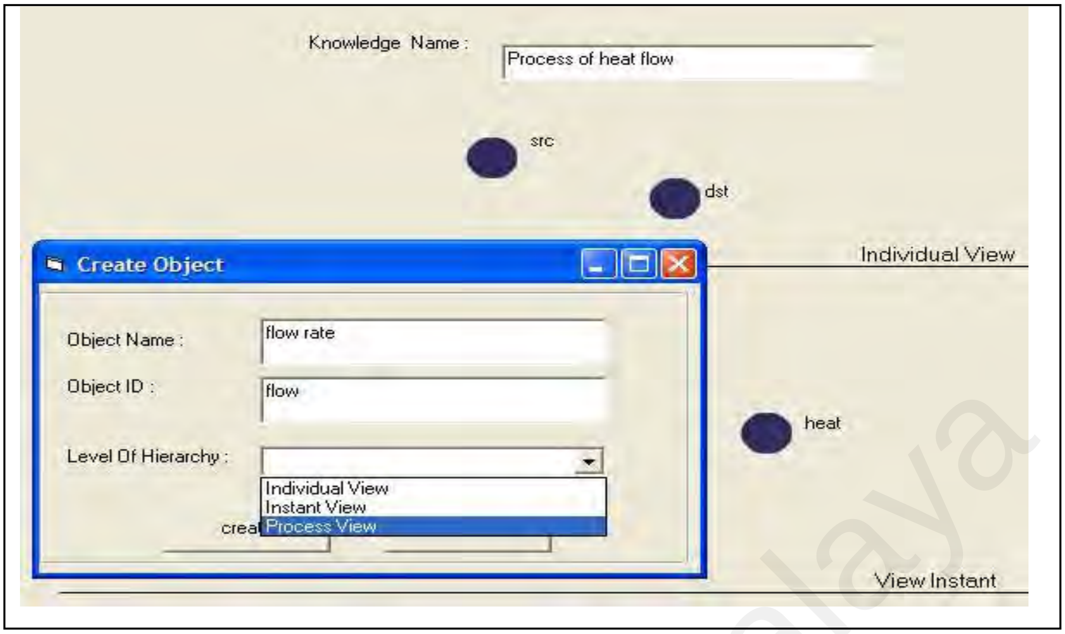


Figure 4.9 :
Window for Declaring Object

Subsequently, the preconditions for some objects need to be specified. A Pre-condition of an object is a constraint where the object must fulfill. The expert needs to specify these pre-conditions in the pre-condition window as shown in figure 4.10.

In the example of heat transfer, we declare precondition for *heat* object where the condition is that the heat of path must be aligned. This means that if the heat path is not aligned the heat transfer between the two objects might not be possible.

Figure 4.10 :
Pre-condition Window

The next stage is to declare the quantity condition for the object. Quantity condition is the initial state of an object before the process occurs. In our example, quantity condition of the object is that one of the object either *dst (destination)* or *src (source)* must have a higher temperature than the other one. It may be the temperature of *scr* is higher than the temperature of *dst* or vice versa. The expert needs to declare this condition in the quantity condition window as shown in figure 4.11:

Figure 4.11 :
Window For Declaring Quantity Condition

The most important activity in this process is declaring the relationship between objects. The relationships will be referred to in generating the narrative text. For example, in the heat flow process, one of the relationships we need to establish is the relation between flow rate and the quantity of different temperatures between *src* and *dst*.

This relation represents how the flow rate influences the heat transfer from one object to another object. When this relation is established or defined, a line between these objects will be established to represent that relation between them. The expert needs to establish it at the relationship window as shown in figure 4.12:

Influence relationship between objects might occur in two ways, either direct or indirect. Direct influence means an object will give effect to another object directly without any medium. Meanwhile indirect influence means, an object will affect another object through a medium.

Figure 4.12 :
Windows for Declaring Relationship Between Objects

After all the information about the process have been declared, a complete graphical view can be seen as in figure 4.13:

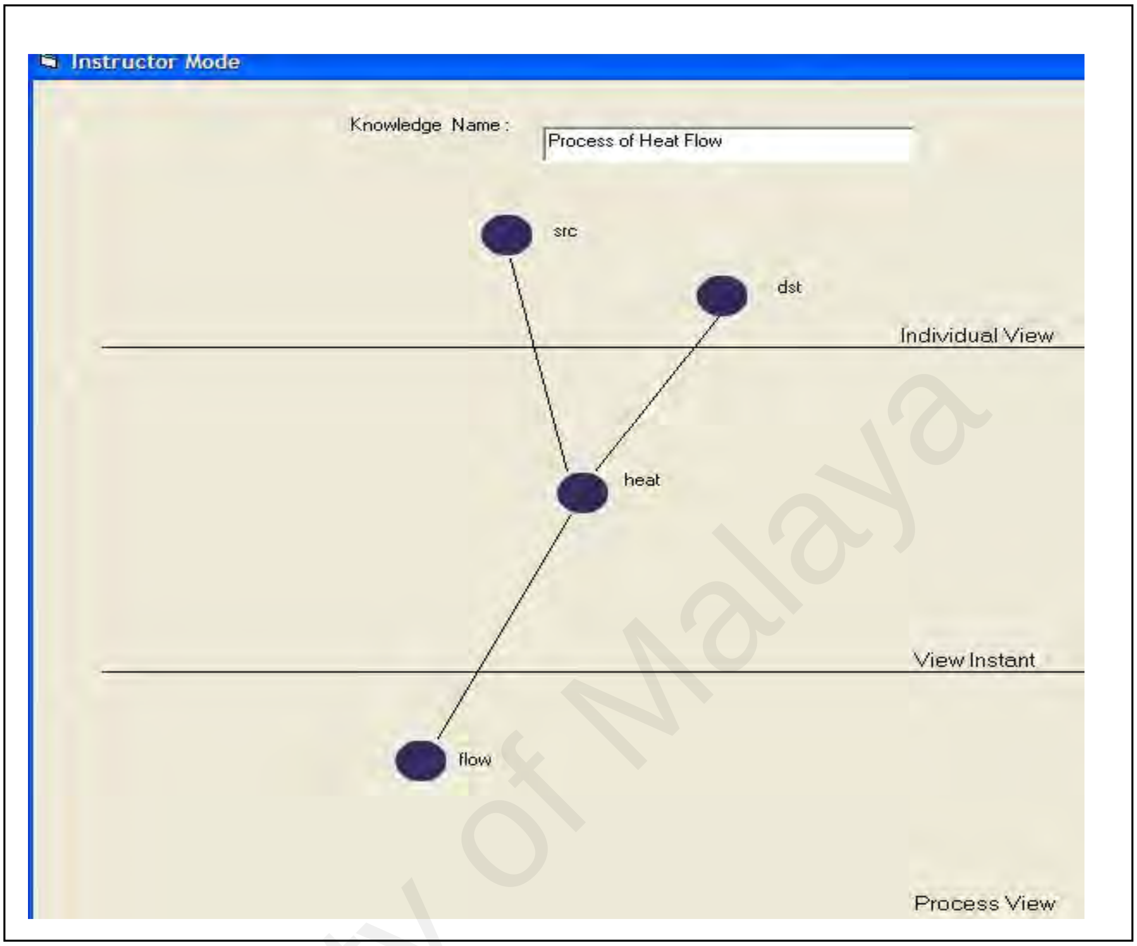
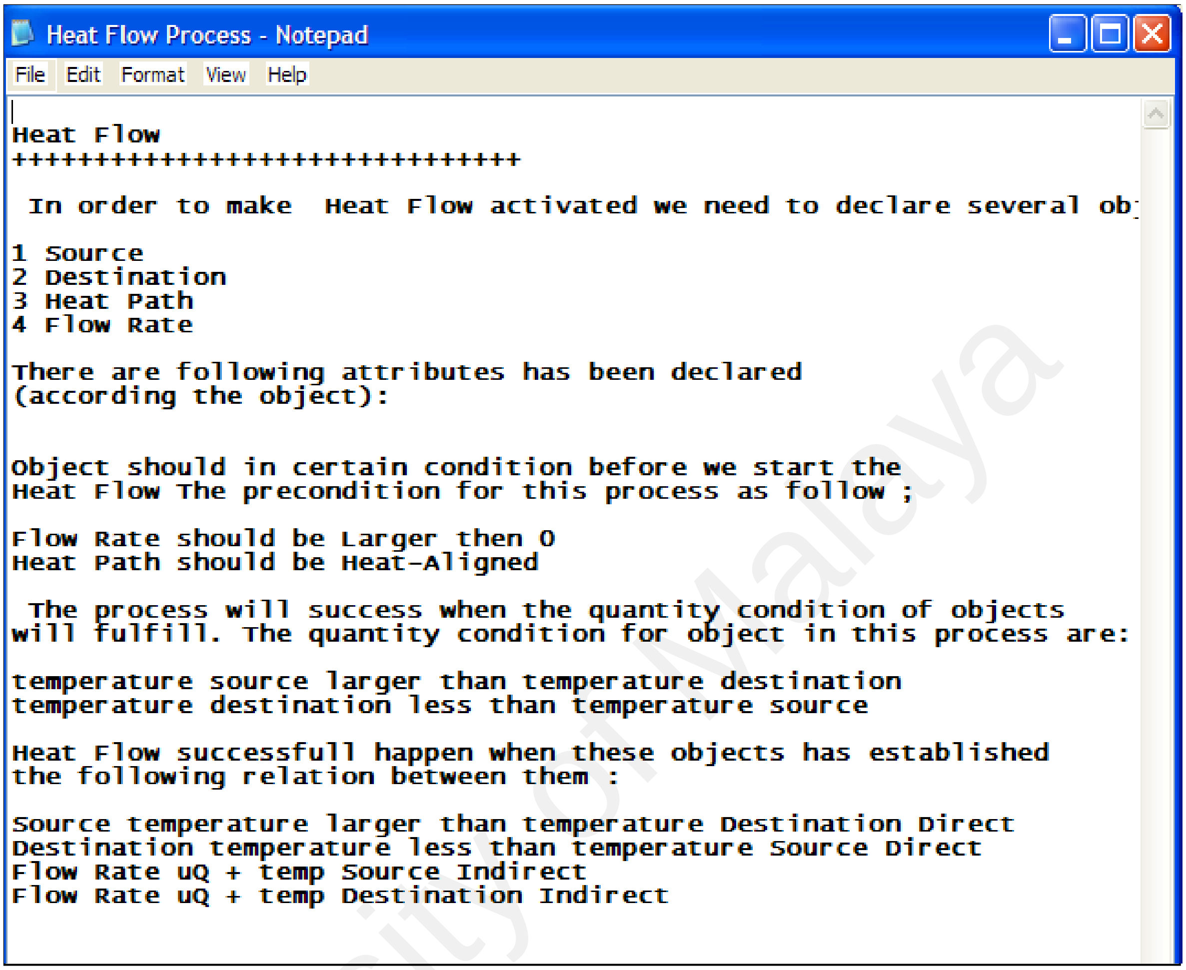


Figure 4.13 :
Graphical View For Heat Flow Process

From this graphical view, a narrative text will be generated. The narrative text form will explain all the information about this process in words and store it in the text format. The text format for this process is shown in figure 4.14 :



Heat Flow
++++++++++++++++++++++++++++++++

In order to make Heat Flow activated we need to declare several ob

1 Source
2 Destination
3 Heat Path
4 Flow Rate

There are following attributes has been declared
(according the object):

Object should in certain condition before we start the
Heat Flow The precondition for this process as follow ;

Flow Rate should be Larger then 0
Heat Path should be Heat-Aligned

The process will success when the quantity condition of objects
will fulfill. The quantity condition for object in this process are:

temperature source larger than temperature destination
temperature destination less than temperature source

Heat Flow successfull happen when these objects has established
the following relation between them :

Source temperature larger than temperature Destination Direct
Destination temperature less than temperature Source Direct
Flow Rate uQ + temp Source Indirect
Flow Rate uQ + temp Destination Indirect

Figure 4.14 :
View of Heat Flow Process.txt

4.4 Conclusion

We have chosen two different domain knowledge to develop chemistry and physics samples of expert modules in this study. For physics we have chosen the heat-flow process and for chemistry we have chosen the precipitate formation of FeS . Several steps need to be followed as a requirement to build this module. The steps are:

- Declaring objects
- Declaring pre-conditions
- Declaring quantity conditions
- Declaring attributes
- Declaring relationships between objects

Two main outputs are the graphical view for the process and the explanation of the processes in the narrative text.

Through the discussions and examples, we have shown how two different expert modules can be generated from one prototype tool. We have generated expert modules for physics domain and chemistry domain as our samples.

Chapter Five

Conclusion and Future Work

5.1 Conclusion

The work presented in this dissertation represents an initial investigation of representing knowledge using QPT approach. Knowledge represented will be used as an expert module in a complete intelligent tutoring system (ITS). For this work, we managed to simulate it using two different domain of knowledge, which are physics and chemistry using a tool prototype that we developed to evaluate this approach.

The main concerns that we are trying to highlight in this work are knowledge representation and the reusability of the product. The Qualitative Process Technique (QPT) approach seems more appropriate in developing expert module as a part of ITS component rather than the conventional techniques such as rule based or semantic network or etc.

Second concern is reusability of the tool and generated knowledge (expert module) itself. Using QPT approach, knowledge will be transformed into a process form. In developing an expert module we are actually developing a small process independently.

We might use this output again and again when we are developing the main expert module for ITS as long as it is relevant to this small expert module that has been produced.

The prototype performs two major tasks, which are preparing knowledge in a graphical view and generating it into a narrative text. In order to perform these tasks, several sub-tasks must be completed first. These include declaring the objects, pre-conditions, quantity conditions, attributes and relationship between objects. After declaring or initializing all the information, the graphical view will appear at workspace and the narrative text form will be generated from this information. General workflow for these processes can be seen in figure 5.1.

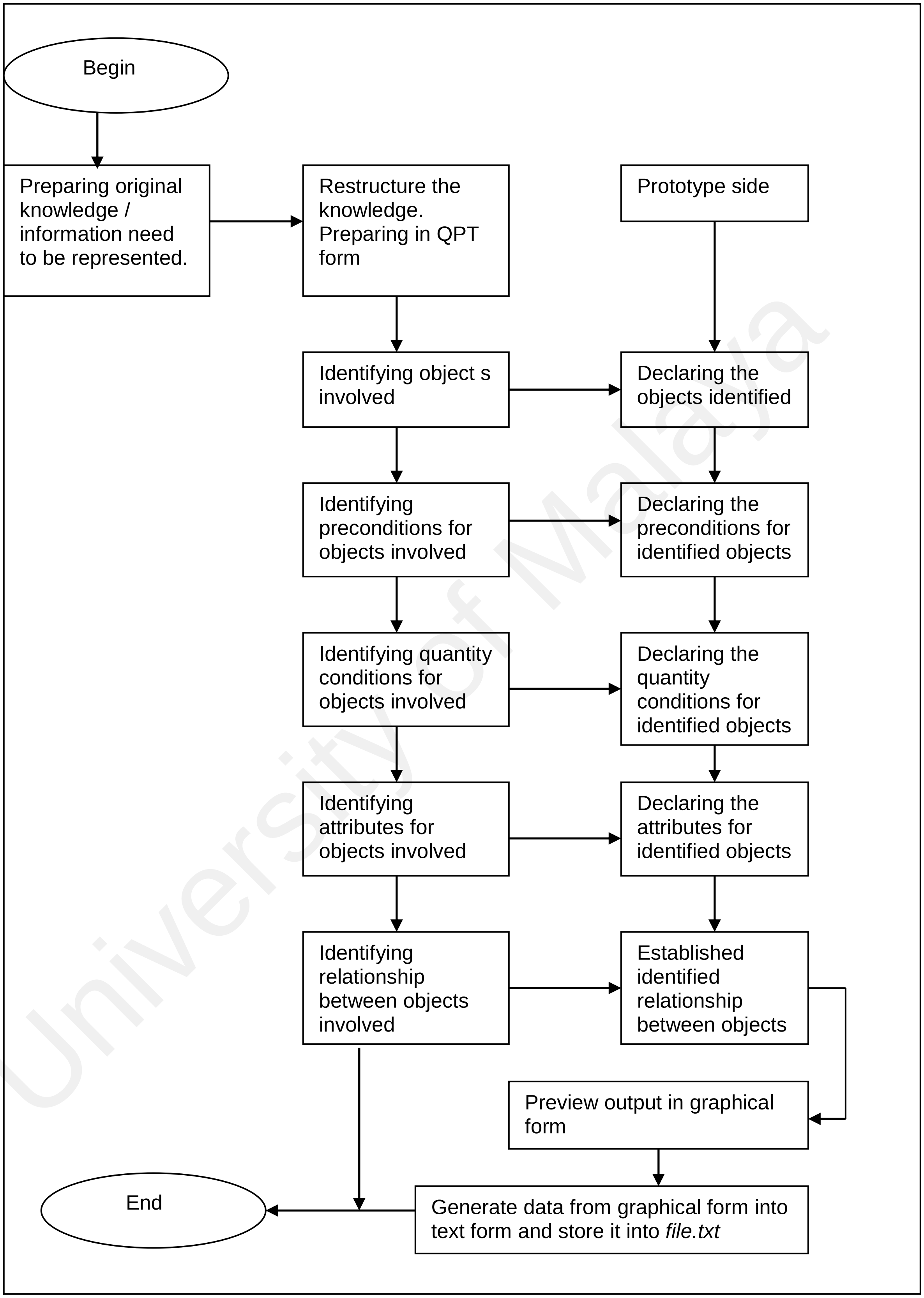


Figure 5.1 :
A Work Flow For Developing Expert Module Using QPT Approach

In conclusion, QPT approach has the potential to become one of the alternative ways to represent knowledge in developing expert module as an ITS component. QPT presents knowledge or information as a process, thus most knowledge or information which deals with processes is very suitable to be represented using this approach.

However, more research or studies need to be conducted to identify better ways to represent knowledge to facilitate better understanding. In fact, researchers in AI community have started with several projects in this area to identify the best way to manipulate QPT in representing knowledge.

5.3 Future Work

Although this work constitutes a significant contribution towards knowledge representation and the reusability of the product, there are many directions in which this work could be enhanced. The final part of this chapter discusses some of the weaknesses in this study and suggestions for future research.

- The main problem encountered when dealing with QPT is to establish a relationship between objects. It is difficult to differentiate between one relationship with another. Redundancy of relationships always occurs. Usually this phenomenon occurs when we are dealing with a process where the relationships sometimes come from the quantity condition of one object to another object's quantity condition.

- The text that has been generated does not explanation much about the process due to lack of information. Using database could be one of the solutions for this situation.

- Combination of several approaches in representing knowledge can give more information to generate. QPT is suitable to represent knowledge in process form, but when dealing with tasks to perform more explanation about the process, we can use another approach such as rule based.

- Explanation that is generated from a graphical view is only in the narrative form. It is better if it can be generated with animation. It would be easier to be understood by common users.

- An engine of the prototype could not synthesize information received. Another artificial intelligence concept could be applied to this part to enable the engine to learn the information and synthesize and represent it in a proper way.

- In software development side, this prototype is developed independently and not tested to integrate with another components to build a complete ITS. It is because the purpose of this study is to investigate if it is suitable to use QPT in representing knowledge. Therefore, for future work we intend to integrate the output from this work to build a complete ITS

R E F E R E N C E S

[1] D. McArthur, M.Lewis and M.Bishay. The Roles of Artificial Intelligence in Education: Current Progress and Future Prospect. Technical Report RAND DRU-472-NSF, RAND, 1993.

[2] M.Twidale. Redressing The Balance : The Advantages of Informal Evaluation Techniques For Intelligent Learning Environments. *Journal of Artificial Intelligence in Education*, Teridak, M.B : 155-178, 1993.

[3] El-Sheikh. E and J.Sticklen. A Framework for Developing Intelligent Tutoring System Incorporating Reusability. In *11th International Conference on Industrial and Engineering Applications of Artificial Intelligent and Expert System,* Benicassim, Castellon, Spain*,* 1998.

[4] T.Alex Reid. Perspective on Computers in Education: The Promise, The Pain, The Prospect. In *Computers and Teaching in the Humanities (CATH) Conference: Courseware in Action.* University of Glasgow, 10th - 12nd September 1994.

[5] J.Beck, M.Stern and E.Haugsjaa. Application of AI in Education. http://www.acm.org/crosroads/xrds3-1/aied.html, Last Modification, 3rd October 1998.

[6] Intelligent Tutoring System, http://www.ott.navy.mil/1-2/1-2-1/index Last Updated : 27th October 1997

[7] http://ltsc.ieee.org/doc/wg3/interactive_glossary/ITS.htm Last Review : 25th December 2002

[8] J.Self. Theoretical Foundations for Tutoring Systems. Technical Report AAI/AI-ED No. 45 in Journal of Artificial Intelligence in Education, 1(4), 3-14), 1990.

[9] Kinshuk, Tretiakov. A, Hong.H and Patel. A, Human Teacher in Intelligent Tutoring System : A Forgotten Entity !. In *Proceeding IEEE International Conference,* Madison WI, USA, 8th June - 8th August 2001: 227-230, 2001.

[10] Michael B. Twidale. Knowledge Acquisition for Intelligent Tutoring Systems.

[11] Cheang et al. On Automated Grading of Programming Assignment in an Academic Institution. In Journal of Computer & Education, 41 (2003) pg 121-131

[12] J.Self, The Role of Student Models in Learning Environment. Technical Report AAI/AI-ED No. 94 in *Transactions of the Institute of Electronics, Information and Communication Engineers*, E77-D(1), 3-8, 1994.

[13] Stakov et al. On Knowledge Representation in an Intelligent Tutoring System. www.pmfst.hr/~stakov/zn-radovs/2000/INES_2000.pdf . Last Viewed : 20th February 2004

[14] A.Vantaggiato. Some Considerations on Intelligent Tutoring Systems, In *Proceedings of 1st International Conference on Industrial and Engineering Application of Artificial Intelligent and Expert System*. Tullahoma, Tennessee, pg. 1163-1167 Vol : 2, 1988.

[15] Hegarty C .'Statutor: Intelligent Tutoring System?', BILETA '96 *Conference Proceedings, 3 The Journal of Information, Law and Technology (JILT).* 1996.

[16] D. Walmsley. Tutor, Student and Expert – Modelling Interpretation in an ITS for Fact Finding, *In Intelligent Tutorial Systems, IEEE Colloquium*, 12nd May 1998, pg : 10/1 – 10/8, 1998.

[17] Dicheva and Dimitrova. An Approach to Representation and Extraction of Terminological Knowledge in ICALL, *In Journal of Computing & Information Technology CIT*, Volume 6, Number 1 1998, pg 39-52

[18] Sebillot and Danna. Expert Module in Compounds, an ITS for Second Language Learning. *www.irisa.fr/texmex/papers/1993/sebillotSDB93.pdf*. Last Viewed : 20th February 2004

[19] Kishuk. Intelligent Tutoring Systems – A New Approach to Teaching . *http://fims- www.massey.ac.nz/~kishuk/thesis/chap3.pdf*. Last Viewed : 15th Feb 2004

[20] Kenneth D. Forbus. 1986. Qualitative Process Theory. *Artificial Intelligence* 24: 85-168

[21] Devedzic and Debenham. An ITS for Teaching Formal Language, *Lecturer Notes in Computer Science*. Volume 1452, Issue 1452 pg 514-523, 1998

[22] Maria Smolarova and Pavol Navrat. *Software Reuse : Principles, Patterns, Prospects*, http://citeseer.ist.psu.edu/357780.html, Last Viewed : 25th May 2004

[23] Ruben Priento-Diaz. Making Software Reuse Work : An Implementation Model. *In ACM SIGSOFT Software Engineering Notes*. Volume 16, Number 3, July 1991

[24] Murray, T. Authoring intelligent tutoring systems: An analysis of the state of the art. *In International journal of artificial intelligence in education*, 10, 98-129, 1999

[25] Salles and Bredeweg, A Case Study of Collaborative Modeling : Building Qualitative Model in Ecology. *Artificial Intelligent in Education: Shipping the Future of Learning Through Information Technology*. IOS-Press/Ohmsha, Osaka, Japan, pg 245-252, 2003.

[26] Ruben Priento-Diaz and Peter Freeman. Classifying Software for Reusability. *In IEEE Software*, January 4(1):pg 6-16, 1989

[27] Sommerville, Ian. *Software Engineering Fifth Edition*, Addison-Wesley, 1998

[28] Pressman, S. Roger. *Software Engineering : A Practitioner Approach*, Mc-Graw Hill, 2000

[29] Muchado and Bredeweg. B, Towards Interactive Tools For Constructing Articulate Simulation, *In Proceeding of The International Workshop Qualitative Reasoning, QR 01*, pg 98-104, San Antonio, Texas, May 17-19 ,2001

[30] Ruben Prieto-Diaz, Software Reuse: Issues and Experiences, *The American Programmer Journal*, May 10(3) p 61-66, 1993

[31] Richard B.Kieburtz and Laura McKinney. A Software Engineering Experiment in Software Component Generation, *In Proceeding of ICSE-18*, 1996

[32] Even-Andre Karlsson, Sivert Sorumgard and Eirik Tryggeseth. Classification of Object-Oriented Components For Reuse. *In Proceeding TOOLS'7 Dortmind*, Prentice-Hall, 1992

[33] Becker and Blois, Consideration On The Applications Of Object-Oriented Reused Technology To The Computer Supported Co-Operative Learning Domain, *In Seventh International Workshop on Groupware (CRIWG '01)*, Darmstadt, Germany, September 6-8, 2001

[34] Frakes et al, Software Reuse : Is It Delivering? , *Proceeding Of The 13th International Conference On Software Engineering*, Austin, Texas US, May 13-17, 1991

[35] Percini, Mecella, Batini, Conceptual Modelling and Software Component Reuse: Towards the Unification, *In Information Systems Engineering: State of the Art and Research Themes,* London, 2002

[36] Ted J.Biggerstaff, Reuse Technologies And Their Niches, *In International Conference on Software Engineering, Proceeding of the 21st International Conference on Software Engineering* , pg 613-614, 1999

[37] Forbus. K, Qualitative Physics : Past, Present and Future, *In Exploring Artificial Intelligent*, Morgan – Koffman Publication, 1988

[38] Salles. P , Deriving Explanation From Qualitative Models, *In Proceeding Of The 8th World Conference On Artificial Intelligent In Education,* Japan, August 18-22, 1997

[39] Roschelle. J et al, Developing Education Software Components, *In Computer Magazine*, Volume 32, Issue 9, ISBN 0018-9162, Page 50-58, September 1999

[40] El-Sheikh, E. Development of a Methodology and Software Shell for the Automatic Generation of Intelligent Tutoring Systems from Existing Generic Task-based Expert Systems. *AAAI-99: Sixteenth National Conference on Artificial Intelligence*, *SIGART/AAAI-99 Doctoral Consortium.* Orlando, Florida, AAAI Press. 1999

[41] Frias and Moore, A Semantic Network For Artificial Intelligent Processing in Tools For AI, *In IEEE International Workshop* on 23 – 25 October 1989

[42] Adler R.M . Emerging Standard for Component Software, Computer Magazines, Volume 28, Issue 3, pp 68-77 , March 1995.

[43] Basili V.R, Briand L.C and Thomas W.M, Domain Analysis for the Reuse of Software Development Experiences, *In Proceedings of the 19th Annual Software Engineering Workshop*, NASA / Goddard Space Flight Center, Greenbelt, Maryland, 30th November – 1st December 1994.

[44] B. Chandrasekaran, R. Josephson, Benjamins. What Are Ontologies, and Why Do We Need Them?, *Intelligent System, IEEE*, Volume 14, Issue 1, pg 20-26 ,Jan/Feb 1999

[45] O'Leary, Using AI in Knowledge Management : Knowledge Bases & Ontologies*, In Intelligent System, IEEE*, Volume 13, Issue 3 , pg 34-39, May / June 1998

[46] Ruben Prieto-Diaz, A Faceted Approach to Building Ontologies, *In 2003 IEEE International Conference On Information Reuse and Integration,* Las Vegas, October 27-29, 2003

[47] McArthur.K, Saiedian.H and Zand.M . An Evaluation of the Impact of Component-Based Architectures on Software, *Information and Software Technology Magazine,* Volume 44, Issue 6, pp 351-359, 15th April 2002.

[48] *Kim, H. ,Qualitative reasoning about fluids and mechanics. Ph.D. dissertation and ILS Technical Report, Northwestern University, 1993*

[49] Ruben Prieto-Diaz, Reuse as a New Paradigm for Software Development, In Proceedings of the International Workshop on Software Reuse. Marjan Sarshar, Liverpool, UK. 8th – 9th January, 1996.

[50] Kruger.C, Software Reuse, *In ACM Computing Surver*, Volume 24, Number 2, June 1992

[51] Frakes W.B, Success Factors of Systematic Reuse, *IEEE Software*, 1994

[52] Mayer B. , The Reusable Challenge, *Computer Magazine*, Volume 29, Issue 2, Page 76-78, February 1996

[53] Tam et al, WITS: A Reusable Architecture for a VR-Based ITS, *In Intelligent Tutoring System 96 Workshop on Architecture, Method For Designing Cost-Effective & ReusableITS*, Montreal, June 10th , 1996

[54] El-Sheikh and Sticklen. J, Generating ITS From Reusable Components And Knowledge-Based System, *In ITS 2002, LNCS 2363*, page 199-207, Spain, June 2002

[55] El-Sheikh and Sticklen. J, Leverage A Task Specific Approach For ITS Generation, In Computing Generic Task & KADS Framework, http://islnotes.cps.msu.edu/domsite/pubpress/isl pubs.nsf , Last Viewed : 2nd January 2004

[56] Kinshuk http://massey.ac.nz/~kinshuk/thesis/chap3.pdf Intelligent Tutoring System – A New Approach To Teaching. Last Viewed : 15th February 2004

[57] *http://burks.bton.ac.uk/burks/foldoc/39/7.htm*, Artificial Intelligent. Last Viewed : 12th Nov 2003

[58] *http://www.cirl.uoregon.edu/research/kr.html*, Knowledge representation, Last Viewed : 12th Nov 2003

[59] *http://www.int.gu.edu.au/courses/2003int/L_0212.ppt*, Knowledge Representation, Last Viewed : 5th January 2004

[60] Jackson. Peter, *Introduction to Expert System* , International Computer Science Series, Addison –Wesley, London, 1999

[61] Iwasaki. Y, Qualitative Reasoning And The Science Of Design, *In Expert IEEE Intelligent System*, Volume 12, Issue 3,page 2-4, May/June 1997

[62] Collins, J. and Forbus, K. Building Qualitative Models Of Thermodynamic Process. *In Proceeding Of The Qualitative Reasoning Workshop.* www.qrg.northwestern.edu/papers/papers.htm#QP_Theory Last Viewed : 15th February 2004

[63] Robert D. Way. www.jsc.nsa.gov/icat/does/blackpaper.html Knowledge Representation, 22nd November 1993.

[64] Bachman and McGuinness, Knowledge Representation, Connectionism, and Conceptual Retrieval, *Association for Computer Machinery*, 1988

[65]*http://www/cs/vassar.edu/faculty/welty/papers/phd/HTML/disseratation/christNelthy*, Knowledge Representation, Last Viewed 14th December 2002

[66] Iwasaki. Y, Real-World Application Of Qualitative Reasoning , *In IEEE Intelligent System*, Volume 12 Issue 3, page 16-21, May/June 1997

[67] Bredeweg. B and Bouwer, VISI-Garp : Graphical Representation of Qualitative Simulation Model, *In AI in Education : AI-ED In The Wired & Wireless Future.* Page 294-305, IOS-Press Ohmsha , Osaka, Japan. 2001

[68] Nussbaum. M and Roses. R, Development Of Intelligent Tutoring System Using Knowledge Structure, *In Computer & Education*, Volume 36, Issue 1, Page 15-32, Elsevier Science Ltd, January 2001

[69] Carver. J, Meta Knowledge : How Knowing What We Know Can Improve Organizational Performance. www.rhadiant.co.uk/resources/meta-knowledge.pdf , June 2002

[70] Forbus, K. (1993). Qualitative process theory: Twelve years after. Artificial Intelligence, 59, 115-123., Last Viewed : 12nd March 2004

[71] Mustapha. SMFD et all, Application Qualitative Process Theory To Qualitative Simulation And Analysis on Inorganic Chemical Reaction , *In Sixteen International Workshop On Qualitative Reasoning QR 2002*, Spain, June 10-12, 2002

[72] B. Bredeweg and K. Forbus. 2003. Qualitative Modeling in Education. *AI Magazine*, Volume 24, Number 4, pages 35-46

[73] H.Salles, P.Salles and B.Bredeweg (2004), Qualitative Reasoning in the Education of Deaf Students : Scientific Education and Acquisition of Portuguese as a Second Language, *In 18th International Workshop on Qualitative Reasoning QR2004*, Northwestern University, Evanston, Illinois, US. August 2-4, 2004