UNIVERSITY
OF TECHNOLOGY
SYDNEY

Integrating Evaluation into Development of Knowledge-Based System for Adaptive E-learning System

by Alva Hendi Muhammad

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Professor Ghassan Beydoun

University of Technology Sydney
Faculty of Engineering and Information Science

April 2020

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Alva Hendi Muhammad declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Information System & Modelling at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:

Signature: Signature removed prior to publication.

Date: 14 April 2020

ACKNOWLEDGEMENT

I would like to express my sincere appreciation to my supervisor, Professor Ghassan Beydoun, for his academic guidance and personal support during my PhD journey. His continued support and patience have encouraged me to complete this thesis. I would also like to thank Professor Iwona Miliszewska from the University of Technology Sydney and Associate Professor Jun Shen from the University of Wollongong, for their support.

I would also express my sincere appreciation to my love, Wirdasari, for her constant and continued support and patience during my journey. My special thanks go to my girls Azkadina and Zivana who have always cheered me and restored my energy. Special thanks also to my boy Ghaza, who was born in Australia during my third year of PhD. I always pray that one day, you all can pursue your study in Australia. I am grateful to my sister, Ayu, for her continuous support. This thesis is dedicated to my beloved father and mother, who always prayed for me and supported me in my PhD journey.

I welcome this opportunity to thank my friends, neighbours and colleagues in Wollongong and Sydney, who have provided me so many memories to share together. In particular, I would like to thank LPDP families from the first generation to the fifth generation: Alfiah, Arya, Anggun, Sujadi, Adit, Mahmud, Giri; PPIA family: Dudi, Karfi, Zul, Wira; and also ICMI warriors: Widhi, Joe, Dhimas, Zulya, Aan.

My gratitude is also extended to the financial assistance provided by the Indonesian Endowment Fund for Education (LPDP), Ministry of Finance Indonesia and the University of Technology Sydney towards the completion of this thesis.

PUBLICATIONS

Muhammad, A., Zhou, Q., Beydoun, G., Xu, D. and Shen, J., 2016, May. Learning path adaptation in online learning systems. In *2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 421-426). IEEE.

Muhammad, A., Shen, J., Beydoun, G. and Xu, D., 2016, July. SBAR: A framework to support learning path adaptation in mobile learning. In *International Conference on Frontier Computing* (pp. 655-665). Springer, Singapore.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

One of the main obstacles in knowledge-based systems, particularly in the field of evaluation, is to determine when the development of a knowledge base is complete with valid knowledge available to use. Ironically, this question is not easy to answer, since it involves many aspects in the lifecycle of knowledge-based systems. Some efforts have been introduced in literature, although none have been widely used or even agreed upon. In our investigation, we found that it is possible to provide a ready to use knowledge base without a separate testing phase. This thesis examines the way in which the testing phase could be integrated into the development phase through a statistical monitoring task during the knowledge acquisition process. This research tailored the dynamic evaluation framework and utilised the RDR technology as an incremental knowledge acquisition methodology, with additional statistical analysis to monitor the development of the knowledge base.

Using this framework, this thesis presents INCAES, an incremental adaptive e-learning system. The knowledge base was developed incrementally for suggesting learning contents to students. The recommendation was provided by analysing the learner model as a representation of the students in adaptive learning. However, a systematic understanding of attributes of learners is still lacking in the field of adaptive learning, since those attributes are regularly changed and updated. Thus, we keep track by investigating the potential learner's attributes that could be adapted to the adaptive system from a number of recent works.

The evaluation of the theoretical concept has been conducted using two real world case studies. The first case study was applied on the Web Programming subject, while the second case study was applied on the Networking Essentials subject. The test cases generated from learners' attributes and values were used to evaluate the knowledge base. Different scenarios were provided for the case study. The first scenario dealt with stopping the progress of the knowledge acquisition process at an early stage, while the

second scenario dealt with stopping it at a later stage. Following the knowledge base construction, two steps of testing were conducted to measure the performance of the initial knowledge base and the subsequent test set. The results have shown that the framework allows for the rapid development of a knowledge base with valid knowledge inside.

1 INTRODUCTION

Knowledge-Based Systems (KBS) have been attracting a considerable amount of interest for many years. It is one of the most successful areas of ongoing research within the field of Artificial Intelligence (AI). Many applications of KBS have been studied covering a wide range of functional areas and problem domains. Among the study of KBS, evaluation is often the forgotten step when developing KBS. Many researchers in the area of KBS evaluation are focused mainly on verification and validation issues. Most of the discussions are usually concentrated on completeness and consistency, error detection and correction, knowledge representation, and maintenance. Ideally, a KBS should be accurate and contain enough knowledge to be of use in the real world. In fact, the long process of development and evaluation to construct an optimal and accurate KBS is always difficult and expensive. This thesis fills a gap in the field of evaluation of knowledge-based systems by utilising a dynamic evaluation approach. Instead of building a complete system and testing it, this research suggests monitoring the knowledge base development process and statistically analysing the performance evaluation of the knowledge base. The thesis does this in the context of dynamic e-learning, an area of substantial significance nowadays.

This chapter has been divided into five sections. Section 1.1 presents the background and motivation of the thesis. The following Section 1.2 describes an overview of adaptive e-learning systems. Section 1.3 sets out the objectives of the research and its significance. Section 1.4 describes the structure of the thesis. Finally, this chapter concludes with a summary in Section 1.5.

1.1 Background and Motivation

Research on the development of the knowledge-based systems has been extensively investigated for many years. KBS is a class of AI systems in which the knowledge of experts is transferred into a computer system to support the decision mechanisms. The scope of a given KBS typically includes problem-solving or decision-making for routine tasks in a particular domain of knowledge. Hence, KBSs have several forms, such as expert systems, advisor systems, decision support systems, or diagnostic systems (Liao 2005; Plant & Gamble 2003). KBSs may also vary with respect to their techniques or approaches. For example, rule-based systems and case-based reasoning systems encode expert knowledge as rules and cases. Even though various forms of KBSs are found, these systems have common characteristics. The construction of a KBS typically relies on a human expert to be consulted for providing an answer or clarifying an uncertainty to a specific problem domain. Over the years, KBSs have been successfully implemented in many fields where most of the applications have been for business-oriented organisations. The distribution of KBSs among industries indicates their visibility in many industries e.g. for accounting, banking and financial services, manufacturing, medical, education, and military (Liao 2005; Wagner 2017).

The design of a knowledge-based system typically includes two main components (Sajja & Akerkar 2010). As shown in Figure 1.1, these components are *knowledge base* (KB) and *inference engine*. The knowledge base is basically a repository of knowledge. It contains information on an expert's knowledge or solution of problems in a particular domain. This information can be collected without concern on the detail of how and when that information should be applied. A KBS utilises a control mechanism process called an inference engine. The inference engine interacts with a knowledge base to deduce insights when taking conclusions. The inference engine can be developed afterwards to derive new knowledge from a given fact. This allows a KBS to answer questions or solve specific problems. The inference engine is responsible for representing the elicited knowledge using several reasoning systems such as logical programming, rule engines, and deductive classifiers. This process is analogous to the role of the human brain in that, even though the information (KB) is continually added and modified, the control process (inference engine) remains unchanged. Therefore, a knowledge base and an inference engine have been common features of a KBS. They allow a KBS to replicate human experts by providing an explanation for every decision made by system. This architecture also can give a KBS the capability to learn new facts into their knowledge base. A critical component of a KBS is also the user interface that enables users to query and interact with the system. It also delivers the output of a KBS, which is a conclusion from a given set of facts.

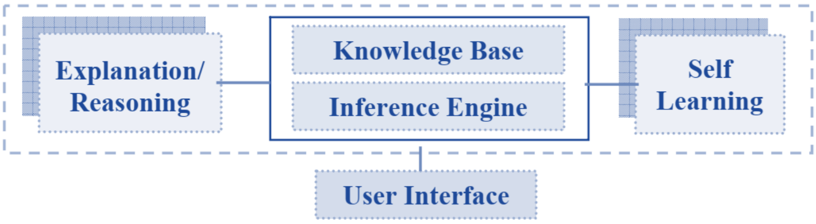


Figure 1.1 Architecture of a KBS (Sajja & Akerkar 2010, p. 3)

According to the knowledge-based system development lifecycle (Adelman & Riedel 1997; Salle & Medsker 1990), the development of a KBS begins with the requirement phase, and is then followed by the knowledge acquisition (KA) phase. Knowledge acquisition plays an important role in the lifecycle to represent knowledge explicitly in the system. The knowledge in a KB has to be very well-defined in representing the problem domain. Knowledge engineers analyse the problem domain or issues in a specific application, and then they represent facts about it by collecting data or information from the domain experts or other sources of knowledge. Several methods have been used to accomplish the knowledge acquisition process such as interviews and discussions with the experts, observing the work environment of the experts, collecting all possible cases and identifying patterns of the process and knowledge. A KBS represents the knowledge by several methods, including frames, rules, cognitive maps and ontologies. The next phases of the lifecycle are design and implementation in which the knowledge is added and stored in the knowledge base. The final phase is evaluation, which is to figure out whether the developed KBS has been completed and is able to reliably produce the required conclusions.

KBS development lifecycle has a number of challenges throughout. A major challenge stems from the knowledge acquisition process itself (Beydoun & Hoffmann 1997a; Compton et al. 1992). The mechanism to capture knowledge from experts has never been an easy task. It is a time-consuming and human-intensive process, and often becomes a leading cause of bottleneck in KA (Beydoun & Hoffmann 2000a). In the past, the experts used to interview and conducted protocol analysis from the psychological field to observe the problem domain. Then, they automated the process using a computer program for efficacy. Both manual and computer-based techniques were significantly efficient at one point, depending on the problem domains and scenarios.

The next issue in KBS is related to the development lifecycle itself. The long period of implementation and the difficulty to critically validate the outputs, have caused the KBS projects to never get past the prototyping stage (Adelman & Riedel 1997). The process is often repetitive, and every iteration must go through another evaluation stage called verification and validation (V&V) to guarantee the reliability of the KBS (Beydoun et al. 2005). Verification aims to address engineering issues related to the required design specifications. This is accomplished by checking the system anomalies and behaviour in case of unexpected inputs. Validation is more focused on results and usage. It ensures that the output of the KB is correct and acceptable for the user. Unfortunately, the purpose of both verification and validation in KBSs has gone beyond evaluation. Some authors (Barr 1999; O'Keefe & O'Leary 1993; Vermesan & Coenen 2013) agree that the evaluation stage must also include refinements to detect and eliminate errors in the knowledge base. While others (Batarseh & Gonzalez 2015; Knauf, Gonzalez & Abel 2002) concentrate on correcting the knowledge base, which is a considerably difficult task. Although extensive research has been carried out in the field of evaluation methodology, studies indicate that KBSs still lack a rigorous evaluation methodology.

Our work in this thesis studies the significant methods and focuses on fast evaluation for delivering a ready to use KBS, by observing its development process. In particular, we develop the idea of dynamic evaluation that was put forward in (Beydoun & Hoffmann 2013). We introduce and evaluate an operational method that tightly couples to the development process in Ripple-Down Rules knowledge base (RDR KB). The method is based on monitoring the knowledge acquisition process and analysing the likelihood of errors that might occur in the future. Furthermore, RDR is a form of knowledge acquisition technique where the knowledge is incrementally built while it is already in use. The advantages of RDR lie in its simplicity, the fact that it provides an ease of use mechanism to maintain an evolving knowledge base, and also the fact that RDR can easily achieve high-level performance with fewer cases to provide (Compton et al. 1992). RDR is currently applied in various applications including pathology, document classification, information retrieval, monitoring system, internet security, and football simulation (Cho & Richards 2004; Finlayson & Compton 2004; Kim et al. 2018;

Nguyen, Nguyen & Pham 2014; Richards 2009).

RDR has its own life cycle, which is different from conventional approaches. RDR does not differentiate the initial development phase and the maintenance phase. The nature of dynamic evaluation makes it particularly suitable for RDR knowledge bases. Typically, most of the RDR evaluations are performed by measuring the accuracy or error level of the knowledge base after the system is completed. The method developed in this thesis can be done simultaneously with system development while doing knowledge acquisition. For RDR development, the result will be that one process will be used where the validated knowledge is incrementally acquired and evaluated. The output of the process will be a validated structure of the knowledge base. As a result of this process, statistical monitoring could gauge the performance of KBs as they grow larger. This is to determine when the evaluation process completes and delivers a ready to use knowledge base.

1.2 Introduction to Adaptive E-learning System (AES)

Our research in KBS evaluation is implemented for the actual domain knowledge in the application of Adaptive E-learning System. According to Collins Dictionary, adaptive means "*having the ability or tendency to adapt to different situations*". Oxford Learner's Dictionaries states that adaptive is "*concerned with changing*" or "*able to change when ... deal with different situations*". This term implies a change in condition meant to deal with different situations. In the field of technology-enhanced learning, various names of adaptive e-learning systems can be found, such as, "adaptive learning environment" and "adaptive learning system". In one of the earliest studies in this field, Brusilovsky (2001) used several terms like "adaptive hypermedia", "adaptive educational hypermedia system", and "adaptive and intelligent web-based educational systems" (Brusilovsky & Peylo 2003). De Bra, Aroyo & Cristea (2004) also suggested the term "Adaptive Web-Based Educational Hypermedia". While a variety of definitions have been suggested, this thesis will use the term adaptive e-learning system as a common term to describe an adaptive and personalised learning system.

According to Kardan, Aziz & Shahpasand (2015), adaptive e-learning systems refer to "*interactive systems that adjust ... their services to users interests, knowledge,* *and goals*". Another definition for this term suggested by Grubišić, Stankov & Žitko (2015) is to "*adapt the selection and presentation of educational content according to the student's status, needs, style, prior knowledge and preferences*". These definitions and some others are emerging and still evolving. However, researchers highlight that AES is a process of adaptation that will provide the best combination of learners' properties (*learner model*) with learning resources (*domain model*) as the services. The services and functionality in AESs may be available in various learning formats, forms, and presentations. The interaction between learner and the learning resources is defined in an instructional model. In essence, instructional model is the core of the expertise from the instructor. AES uses instructional model to define rules and strategy of adaptation. Figure 1.2 illustrates the relation of the three components and the process of adaptation in AES.

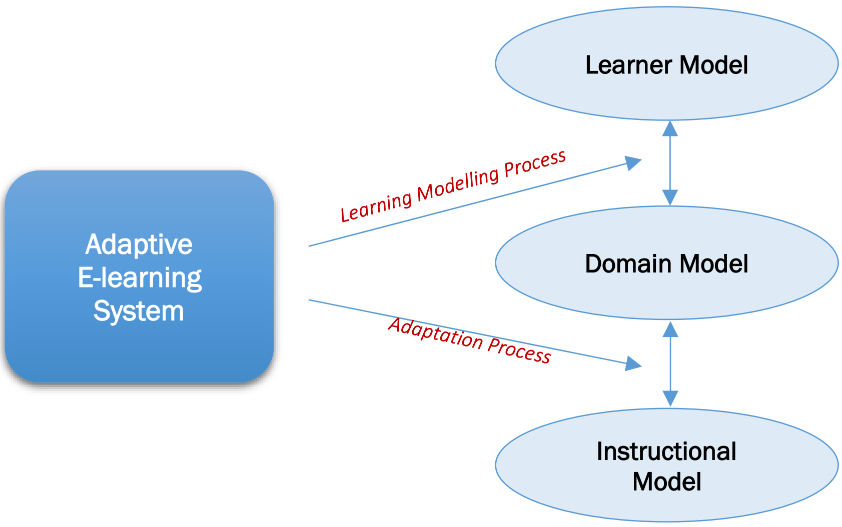


Figure 1.2 The Process of Adaptation in Adaptive E-learning System

AES aims to support and enhance the ways students learn in a formal and informal e-learning system. For many years, e-learning has been designed only for storing educational materials, delivering it to students, and sometimes assisting interaction between teachers and students. The factory model e-learning only meets the needs of average students and no longer meets the needs of each individual. Together with the traditional educational system, e-learning has followed a *one-size-fits-all* approach, which provides identical learning materials to everyone with different background and knowledge. Thus far, many researchers have been concentrating their efforts to make an appropriate alteration to the specific learner by investigating the potential of adaptive e-learning systems (Atif, Benlamri & Berri 2003; Brusilovsky 2001; Canales et al. 2007; Klašnja-Milićević et al. 2011; Papanikolaou et al. 2003; Shute & Towle 2003). This concept has attracted increased interest since AES has complex dimensions, from modelling the learner, to a mechanism for searching information for providing relevant learning resources.

Recent developments in AES have heightened the need for an effective and engaging online learning experience. In their recent research report, the Gonski Institute for Education (2018) recommended the Australian society to prepare people for future education, which is *adaptive* and *personalised.* In their recommendation, the major shift in education has transformed education into "*personalised learning that will involve new interactive and adaptive learning environments*" with the increased use of AI and robotics. The National Academy of Education (2013), which has undertaken research studies addressing educational issues in the United States, has considered to implement the adaptive systems for education. In their report, adaptive learning can be used to improve individual learning experiences by considering several individual characteristics, such as preference, interests, traits, and prior achievements. Additionally, the National Academy of Engineering (2014), an international group of leading technological thinkers from people around the world, has placed *advance personalised learning* as one of their 14 grand challenges for engineering in the 21st Century.

In this thesis, we presented a new paradigm to an initial question in the growing area of adaptive learning, which is 'to what extent the system should be adapted?' and 'what characteristics of the learner to make a pleasant learning experience?' We have reported a review study to complement existing literature studies that acknowledge the role of adaptive learning system in the growing area of dynamic e-learning (Muhammad et al. 2016). We also suggested a framework to address the abstract model of adaptation process in online learning by analysing the student's situation and background (Muhammad et al. 2018). Basically, the implementation of adaptive learning heavily relies on the processing of analytical data and utilising the artificial intelligence for recommendation. Starting from the framework as an initial concept, we extended it with a novel adaptive e-learning system that does not relies on analytical data. Rather, in this thesis we developed an incremental construction of a rule-based system with the guidance of an educational expert for deciding the possible learning paths.

1.3 Research Issues

KBSs have been successfully applied to a wide range of problem domains such as diagnosis, planning and scheduling, control and monitoring, classification, design, prediction and interpretation (Wagner 2017). The major advantage of KBSs lies behind their simplicity and efficacy in design and development (Sajja & Akerkar 2010). However, there are several challenges associated with developing KBSs, from knowledge acquisition bottlenecks to evaluation stages that never complete or provide adequate and timely confidence level. Specifically, the main research question addressed in this thesis is *how to provide rapid analysis and to deliver a valid knowledge base in a timely manner?*

The thesis investigates this problem in the important domain, adaptive learning. It presents two actual case studies for further investigation in adaptive learning illustrating and refining the proposed approach. Two adaptive learning systems are designed to tailor the learning content to the needs of individual learners. Different from other evaluation methodologies, this thesis operationalises and develops a dynamic evaluation approach for validating a KBS. The idea of monitoring the knowledge base development process instead of evaluating at the end of the development process was suggested but never operationalised (Beydoun & Hoffmann 2013). This thesis is aims to unify the stages of development, maintenance, and evaluation simultaneously, into one stage. This method facilitates the development of a knowledge base without needing a separate evaluation phase. During the knowledge acquisition process, the system undergoes an addition and refinement of rules. Rather than waiting for the knowledge base to be completed, this method suggests monitoring the changes during the knowledge acquisition process and concurrently evaluates the expected performance of the knowledge base. This mechanism is to used determine the condition when rules are added to a knowledge base produce no significant effects on the performance of the knowledge base as a whole. When the expected performance has been reached, the evaluation is practically complete at a certain degree of confidence. Therefore, this is expected to cut the time needed to develop a KB since the evaluation does not need to be done waiting for all the data to be tested. Although this idea is promising in theory for improving the evaluation of a knowledge base, this thesis operationalises it and puts it in practice in the increasingly important domain of *adaptive learning*. The thesis fills the gap between theory and practice. It successfully incorporates the knowledge evaluation requirement into the development process itself of a knowledge base.

In order to achieve the main goals, the objectives of this thesis are scoped as follows:

1. To identify important factors that influence the learner model in an e-learning domain. The relation between the learner model and the domain model will be used to identify a suitable adaptation model for the case studies.
2. To construct a set of test cases which are extracted from the learner model and domain model. Pairwise testing is utilised for generating the test cases. The result of this process is the collection of test cases for testing the knowledge base.
3. To introduce three phases of a dynamic evaluation framework as a method to evaluate the knowledge-based system by monitoring its development status.
4. To develop an RDR-KBS for tailoring the learning contents in an adaptive e-learning environment. The RDR enables the domain expert to incrementally develop the adaptation rules in a structured manner by analysing the test cases.
5. To create suitable AES adaptation models for each case study by examining relations between the learner and domain models and determining relevant factors.
6. To examine and refine the evaluation process during the development of the knowledge base and demonstrating the validity of the resultant knowledge bases.

1.4 Research Significance

The main goal and objectives of this thesis have been presented in the previous section. This thesis proposes several contributions that could be viewed from two perspectives, i.e. theoretical and practical. In the theoretical contribution, this thesis presents an approach to construct a knowledge-based system that supports maintenance and evaluation in one stage. Prior to this study it would take a considerable time to deliver a ready to use KBS. This thesis contributes at improving KBS delivery by evaluating its performance before the KBS is finished. The realisation of this method is in the form of three phases of the dynamic evaluation framework. This framework is initiated from the construction of test cases using pairwise testing and then proceeded with the evaluation of the test cases. A significant part of the new evaluation approach is a statistical analysis module. The final phase is the analysis of evaluations, to allow experts to determine the continuation of the KBS development project. This thesis also proposes a contribution related to the individual learning characteristics or the dimensions of the learning adaptation that are presented in the system. In order to renew interest in the current literature, our analysis has identified the potential dimensions which directly affect the decision to tailor individual needs in adaptive learning environments.

With regards to practical contributions, this thesis proposes a contribution in the form of an incremental-based adaptive e-learning system (INCAES). The proposed system is a rule-based adaptive system with rule sets derived using presented domain cases incrementally. However, this method differs from any existing rule-based systems that requires the experts to declare and programme an explicit rule definition in their system. Instead, our system keeps the complexity of the knowledge base structure hidden from the experts. The authors of the rules are guided to where best to add the rule to maintain the validity of the knowledge base. All rules will be created in a way to ensure consistency of the knowledge base with respect to its past performance. Thus, the set of rules grow incrementally in compliance with the growth of the knowledge base.

1.5 Thesis Structure

This thesis is structured into eight chapters. The activities and descriptions of each chapter are described in the following:

- *Chapter 1 – Introduction*

  The first chapter introduces the thesis with an overview of the background and motivation of the research. Specifically, the problems related to knowledge base evaluations and adaptive learning domains are presented to introduce the purpose of the research. After the context of research is described, the defined goal and objectives to achieve the goal are discussed throughout the thesis. This is followed by the proposed research contributions of this research. The final section describes the structure of the thesis, followed by the chapter summary.

- *Chapter 2 – Literature Review*

  The second chapter investigates recent studies in related literature. In particular, the investigation starts with the domain model, which is the theoretical background related to adaptive e-learning systems. Moreover, it presents the techniques of adaptive e-learning systems. Chapter 2 reviews knowledge-based systems relevant research, with a focus on incremental knowledge acquisition and ripple-down rules. Lastly, this chapter investigates the difficulties in evaluating knowledge-based systems generally and presents some relevant works in this area.

- *Chapter 3 – Research Design*

  This chapter gives an overview of the design science research methodology used toe guide this research. This chapter describes in detail four phases of activity to achieve the goals of this research: (1) problem identification; (2) identifying a generic learner model; (3) development of incremental knowledge and acquisition of the knowledge base; and (4) testing, evaluation and refinement of the concept.

- *Chapter 4 – Identifying the Learner Model of Adaptive E-learning Systems*

  A learner model is synthesized using related literature for adaptive e-learning systems. This is presented in Chapter 4 of this thesis. The chapter identifies and presents the generic learner model to represent an individual student in an adaptive e-learning environment. The characteristics of the learner are important to enable formulation of decisions in adaptive learning. This chapter covers two significant areas i.e. modelling and development of adaptive e-learning systems, to further support the scenarios that are used in this thesis. The design and evaluation of the incremental-based adaptive e-learning system are elaborated in chapter 5 to chapter 7 as the main chapters of this thesis.

- *Chapter 5 – Dynamic Evaluation Method for RDR Knowledge-Based Systems*

  This chapter describes in detail the general concepts and design of the KBS proposed throughout this thesis. The chapter begins with an overview of the validation process in KBSs, followed by presenting the proposed three phases of dynamic evaluation framework for validating a knowledge base. The final section presents the design of the architecture to support incremental knowledge acquisition in a KBS and also demonstration of the implemented system.

- *Chapter 6 – Case Study 1: Web Programming*

  This chapter presents the first case study which is used to observe the evaluation process. The purpose of this case study is to understand the performance of the knowledge base with minimum data presented into the system. Therefore, the expert could take a decision to terminate the project immediately. The chapter illustrates the approach in adaptive learning for tailoring learning contents in the Web Programming subject. In the final section, a discussion of the result and conclusions is presented.

- *Chapter 7 – Case Study 2: Networking Fundamental*

  The second case study is designed to find out how the method works in a different environment. The scenario in the second case study is to take a late decision with considerable data supplied into the system. The experiment was validated by using the different subject of study. This subject has different characteristics for which the lessons are longer with a variety of learning paths. The Networking Fundamental subject is used to instantiate the adaptive learning scenario. The results of the evaluation and its analysis are presented. This is followed by a discussion of the comparison results between two case studies.

- *Chapter 8 – Discussion*

  The purpose of this chapter is to present the results and discuss the findings which emerge from the two case studies. In general, the findings of the two case studies will determine whether the developed KBS could be evaluated immediately as the knowledge grows. This section also includes the performance measurement and analysis.

- *Chapter 9 – Conclusion*

  The last chapter closes the thesis by discussing the summary of this thesis, the main results that contribute to the work of this thesis, and the limitations of the research. Finally, this chapter presents some new topics that require further exploration as future work.

1.6 Chapter Summary

This chapter broadly introduced the background and motivation of this thesis, discussed the research goal, and outlined the thesis structure. Primarily, this thesis addresses the methodological support to evaluate a knowledge-based system. This process was applied to support the modelling of adaptive learning systems and their potential improvement through the development of an incremental knowledge acquisition knowledge base. This chapter also elaborated on the research goals to be achieved and their significance. Finally, this chapter outlined the structure and content of the thesis. In the next chapter, the general concepts and technologies of adaptive e-learning systems are introduced. Knowledge-based systems and ripple-down rules will then be highlighted in general, followed by a discussion of the validation methodology for KBSs.

2 LITERATURE REVIEW

This chapter presents a literature review, which is subdivided into five sections. The first section reviews the literature on the domain model which is the adaptive e-learning system. The main topics covered in section 2.1 are the requirements and challenges, benefits of adaptation, and the conceptual model of adaptive e-learning systems. Following by the section 2.2 which presents the discussion related to the characteristics and classification of learner model. The section 2.3 highlights the general approaches for modelling the adaptive learning system. Existing research related to adaptation and personalisation in e-learning is introduced and discussed. Then, section 2.4 presents a foundational framework of incremental knowledge acquisition that will be implemented in this research. The discussion will be focused on the structure and classification of Ripple-Down Rules. Section 2.4 of this chapter discusses several mechanisms to validate the RDR-KB. The last part of this chapter concludes the literature review.

2.1 Adaptive E-learning Systems

Recent progress in the field of the adaptive systems has led to a renewed interest in the development of adaptive learning technologies which offer students with a remarkable learning experience. Adaptive learning has been found in many learning platforms, from a conventional learning management system, intelligent tutoring system, online courses, open courseware to a revolutionary learning platform, such as massive open online courses. All of these platforms can be designed to tailor the needs of individual students. The broad range of information, from personal details, interests, preferences to environments, can be collected to provide contextual information for students. Researchers have used several methods to provide an initial data as the contextual information for learners. What we know is mostly based upon analytical data-driven. Most of them mined user's data and analysed learning history from past activity.

The common way to compensate for the missing initial data is by using a pre-test or post-test approach to provide a basis dataset for adaptation (Latham, Crockett & McLean 2014; Vesin et al. 2013). Other authors intentionally collected data from previous learning activity in the system as the starting point. Then, a recommendation could be suggested based on the similarity of two learning sessions (Wiedmann et al. 2016), collaborative filtering, or pattern mining (Chen et al. 2014).

Recent studies have shown that dataset gathered from open educational resources (OER) could also be exploited for initial data in an adaptive learning activity (Sun, Cui, Beydoun, Chen, Dong, et al. 2017; Sun, Cui, Beydoun, Chen, Xu, et al. 2017). This approach is designed to assist student in a fragmented time during micro learning activities through online learning. The initial data in this approach is pooled from both online and offline computation (Sun, Cui, Beydoun, Chen, Xu, et al. 2017). The online computation data are retrieved from the real time system, while the offline computation data are gathered from any available OER. Although this approach is technically promising, the implementation will require the pre-processing activities that is segmenting and annotating the learning materials (Lin, Sun, Cui, et al. 2019). The realisation of segmentation and annotation also rely on information about learning content and demographics. In reality, the dataset that provides information about historical learning record are often insufficient or even non-publicly available (Lin, Sun, Shen, et al. 2019). In this thesis, we collected datasets from reputable models and constructed a learner model for adaptation of e-learning system. Thus, our model does not rely on analytical data. Instead, the historical data was used to provide a context for an expert instructor to incrementally develop a rule-based system. The rules provided by the expert instructor become the source of knowledge for the adapting the learning path according to the learning context.

In the next section, we will present our systematic investigation of the various forms and the historical lessons of adaptive learning.

2.1.1 Classifications of AES

Many studies provide a set of features that guide to identify and understand the classifications of adaptive e-learning systems (Brusilovski, Kobsa & Nejdl 2007; Peña-

Ayala 2012). These features, which are always involved when discussing the AES, have turned the elements into a framework for AESs. Although the wide array of features can be used to classify AESs in many forms, there are two basic questions that should be considered to express the functionality of adaptation in an AES (Graf & Kinshuk 2008):

1) *"What can we adapt to?"* This feature expresses the characteristics related to the learner. For example, preferences, previous knowledge, interest, goals, learning styles, cognitive traits, context and environment.

2) *"What can be adapted?"* This feature expresses the services from e-learning systems, such as learning content, presentation or media, and link navigations.

From these general questions, several studies have shown a significant increase in suggesting the research area in AESs. (Manouselis et al. 2014) suggested three research areas in AESs, i.e. user and item data, methods and techniques, and platforms and tools. In (Kardan, Aziz & Shahpasand 2015), they also proposed a simple classification for AES, i.e. adaptive techniques and applications of adaptive techniques. As shown in Figure 2.1, the adaptive techniques provide the adaptation capability for e-learning systems based on various methods. The most widely used techniques were (1) machine learning and soft computing; (2) semantic ontology; and (3) application software. The applications of adaptive technologies addressed the implementation to achieve the adaptation of e-learning. The most applied fields in this classification were (1) learner's problem alleviation; (2) presentation; and (3) learning style detection.

Another detailed analysis by (Gómez et al. 2014) had identified four characteristics of adaptation in terms of learning activities of users. They classified AES as follows:

1. *General adaptation*: This adaptation is found in many forms and typically deals with individual recommendations which are derived from contextual elements or environments of students.

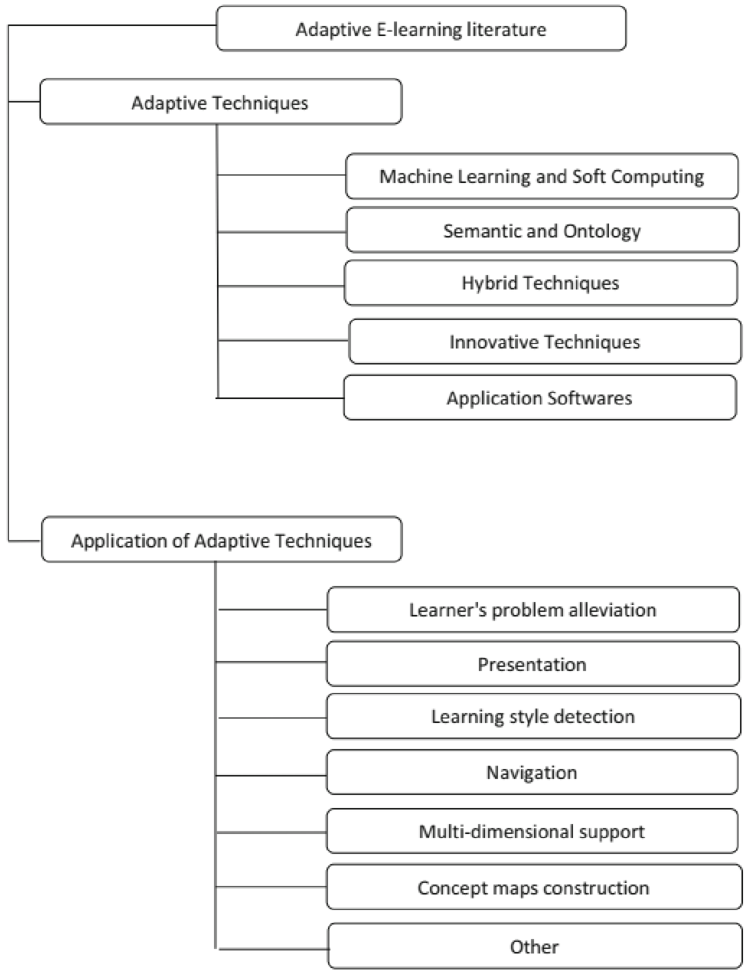


Figure 2.1 AES Classification Framework (Kardan, Aziz & Shahpasand 2015, p. 369)

2. *Feedback and support (scaffolding)*: The main characteristic of this adaptation is assisting the user by providing personal hints and suggesting relevant learning activities based on the activity of students when interacting with the system.
3. *Navigation to locations*: This adaptation provides a learning activity that is relevant to the location and activity in a real-world situation. This adaptation is commonly used during a museum visit or experiment in a laboratory.
4. *Communication and interaction*: This adaptation focuses on improving interaction between students in real life when performing learning activities. The implementation is usually based on a location service or selecting appropriate collaboration tools based on preferences and environment of learners.

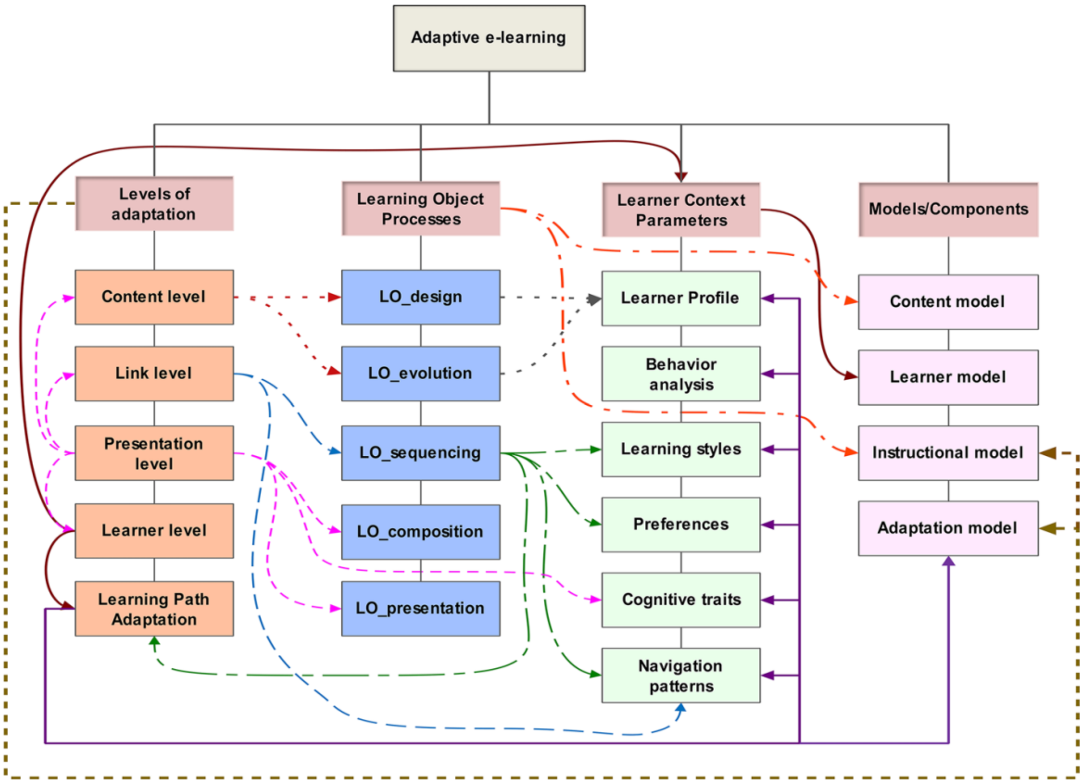


Figure 2.2 A Comprehensive Adaptive E-learning System Classification (Premlatha & Geetha 2015)

Another in-depth review of literature has been conducted by (Premlatha & Geetha 2015). As shown in Figure 2.2, they provide more detailed classifications of AESs by considering their level of adaptation, learning object processes, learning context parameters, and models/components. The classification based on the level of adaptation itself consisted of five levels of adaptation:

1) *Content level adaptation*. This adaptation is applied for learning objects and usually based on personal profile, cognitive level, and learner preferences of the learner.

2) *Link level adaptation*. The main characteristic of this adaptation is sequencing the learning object, by using navigational patterns, features of preferences, instructional design, and ordering the learning path.

3) *Presentation level adaptation*. This level of adaptation considers the composition and presentation of learning objects. The adaptation strategy of presentation is often based on the device and time spent for learning.

4) *Learner level adaptation.* This common adaptation is implemented by considering the context of the learner, for instance, preferences, personal background, behaviour analysis, learning style, and cognitive traits.

5) *Learning path adaptation.* This level of adaptation supports the learner by providing a learning path that should be followed while learning, by considering several characteristics of the learner.

Our investigation has also reported that the process of adaptive learning relies heavily on the parameters of the context. As shown in Figure 2.3, we found that the level of adaptation in may be divided into three domains: learning object, learner, and pedagogical. The main idea of the use of learning objects in e-learning is to support reusability. The common procedure for supporting reusability is by developing educational content into small chunks, so that it can be reused in various learning environments. Furthermore, most adaptive online learning systems consider the learner parameters to customise learning content. The adaptation based on pedagogical preferences is also needed in order to organise learning path in a comprehensive way.

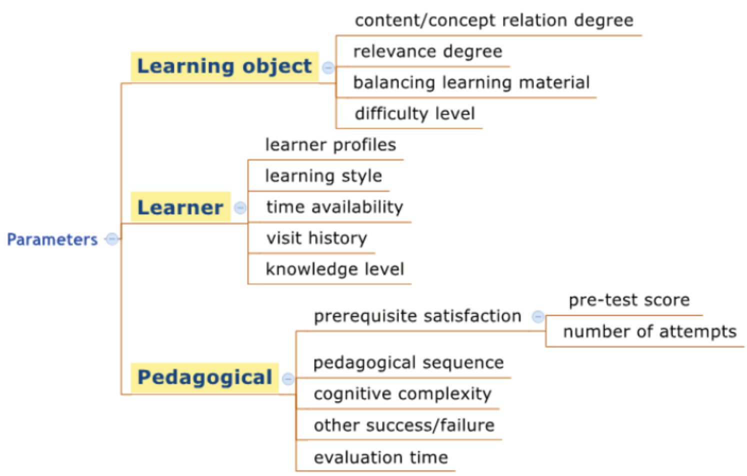


Figure 2.3 Several parameters adaptation in online learning

2.1.2 Potential of Adaptation in Education

Previous research in adaptive learning has been carried out in a wide range of formats and platforms. Adaptation can be formed in the traditional hypermedia systems (Karampiperis & Sampson 2005; Papanikolaou et al. 2003), web-based learning

(Klašnja-Milićević et al. 2011), virtual learning (Xu et al. 2014), mobile learning (Chiang et al. 2016; Gómez et al. 2014), and context-aware ubiquitous learning (Hwang et al. 2010; Yin, Chuang & Hwang 2014). Therefore, many benefits can be obtained by implementing an adaptive e-learning system.

In their review study, (Chiang et al. 2015, 2016) reported that adaptation in mobile learning could lead to an improvement of presentations and contents. It also could stimulate the interests of learners and provide personalised intelligent content for students. The study of adaptive learning on mobile devices has also been reported by (Gómez et al. 2014). They support adaptation of learning activities and educational resources by considering interest, need, preference, device, and physical conditions of learners. The empirical investigations by (Xu et al. 2014) in a virtual learning environment have suggested that adaptation could enhanced effectiveness of examination, improve satisfaction and self-efficacy for learners.

In web-based hypermedia platform, (Xu et al. 2014) reported that adaptation by considering total time spent on each module, number of visited modules, and history of the selection, has guided students to learn at their pace according to knowledge level and progress. In a similar platform, it has been demonstrated that learning style and instructional design could enrich the adaptation functionality for students (Klašnja-Milićević et al. 2011). Previously, adaptive educational hypermedia has facilitated freshman students in high school or college in learning physics (Papadimitriou, Grigoriadou & Gyftodimos 2009). Adaptation through interactive problem solving has been known to improve learning performance and learning activity of the students. (Tarpin-Bernard & Habieb-Mammar 2005) documented the increasing use of hypermedia presentations by modelling the cognitive style that relies on memory, attention, executive functions, language and visual and spatial abilities.

The investigation of AESs to formulate learning path construction have shown the improvement of learning activities through diversity of learning paths (Hwang et al. 2010; Kurilovas, Zilinskiene & Dagiene 2014; Yang, Li & Lau 2010). At the same time, it could also reduce the time spent by students on unnecessary activities while learning. The simulation of learning path selection by (Karampiperis & Sampson 2005) also provided evidence that an accurate learning path could be generated easily by considering learning goals and decision models. Improving adaptation of the content and presentation by considering technological situations such as tools, device characteristics and capabilities, have been reported as one of the benefits of adaptations in context-aware ubiquitous environments (Abech et al. 2016; Benlamri & Zhang 2014). Adaptation through situation-awareness in e-learning systems has revealed the flexibility of learning by providing modularised content in accordance to specific learning scenarios (Pernas et al. 2012). In the same platform of a context-aware ubiquitous learning environment, (Yin, Chuang & Hwang 2014) proposed a hyper-heuristic approach to enhance the quality of student's learning path. They utilised various physical world constraints to plan individual learning paths. The evaluation of these in elementary schools has resulted in the improvement of learning activities.

2.2 Characteristics and Classifications of Learner Model

In recent years, there have been extensive studies that investigate reusable models to facilitate the development of AES. Keeping track with recent studies in this field by revisiting and updating the findings is clearly important, in particular evidence shows that user models in AES change and evolve over time (Granić & Nakić 2010; Nakic, Granic & Glavinic 2015). The constant changes in the field of AES have become an additional motivation for this research to explore the potential learner's attributes that could be adapted to the adaptive system. This section discusses some models and frameworks that characterise the components in AES. The main focus of the discussion is the historical and future aspect of the learner's model to give an idea of the overall implication of the learner with the adaptive system.

2.2.1 Overview of Learner Model Characteristics

One of the central problems in adaptive e-learning is dealing with modelling of the learners, learning resources, and educational learning strategy (Brusilovski, Kobsa & Nejdl 2007; Chrysafiadi & Virvou 2013b; Grubišić, Stankov & Žitko 2015; Peña-Ayala 2012). A learner model contains all definitions and rules for the interpretation of observations about the learner. A learner model also enables more effective analysis of user data and enables interpretation of characteristics of the learner profile. A learner profile essentially contains a data structure that represents the characteristics of a student at a particular moment of time. The data in the profile can be given directly by the user or automatically derived by the system from the behaviour of the user. The process of representing the student in an adaptive learning process is named as *learner modelling*. It is important to identify those concepts that are critical for learning in any given time series describing the learning process. This will in turn-in accordance with the purpose of AES, be the basis to provide the student with a high achievement and pleasant learning experience. Understandably, learner modelling has been one crucial work stream in AES. In practice, it relates to representing, storing, and maintaining some learner characteristics to construct the profile of users to adapt with the learning resources (Brusilovsky & Millán 2007). Figure 2.4 illustrates the basic concept of the learner model and learner profile.

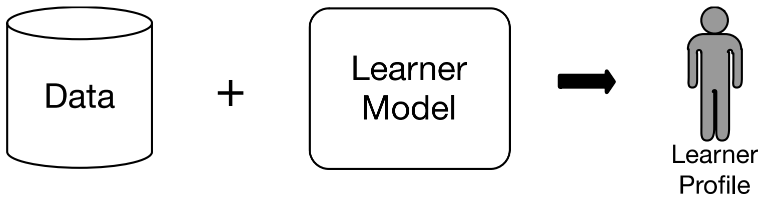


Figure 2.4 Basic concepts of Learner Model

Historically, research on learner models had begun since 1970 from the Intelligent Tutoring System (ITS) domain. Learner modelling is derived from one of the major components in ITS, which is *student modelling*. ITS has utilised student models that contain a set of cognitive, affective, and other psychological states to diagnose student's learning (Burns & Capps 2013; Elsom-Cook 1993). Considering its importance in ITS, it appears that the student model is also popular in the web-based education area. In adaptive hypermedia, the student model is known as *user model*. Moreover, user model has evolved into the main foundation for personalising and adapting the hypermedia (Brusilovsky 2001). In the same view, adaptive e-learning systems employ the learner model not only for storing individual profiles of the learner, but also for tracking the learning style, performance, and other states that evolve during the learning activity (Canales et al. 2007; Jeremić, Jovanović & Gašević 2012).

To construct a learner model in an AES, Paneva (2006) defines a question-based analysis consisting of pursuing answers to the following questions:

- *"Who"* is being modelled?
- *"What"* are the characteristics of the learner?
- *"How"* to model and maintain them?
- *"Why"* should we elicit information from the learner?

Similarly, this question-based approach is also found in (Graf & Kinshuk 2008; Premlatha & Geetha 2015), which identifies features that should be taken into account when providing adaptation. Initially, the learner characteristics presented here began with the investigation of several attributes including age, gender, personality, cognitive abilities, experience, background of learning, knowledge level, motivation, learning style, goals, interest and preference from e-learning.

It is envisaged that age and gender of the learner are usually related to the experience and background knowledge (Brown et al. 2009). These attributes are usually also included in personal information along with language, education level, and occupation (Medina-Medina, Molina-Ortiz & García-Cabrera 2011; Tarpin-Bernard & Habieb-Mammar 2005). However, personal attributes itself are less popular for adaptation when compared to the pedagogical states, such as cognitive abilities, learning style, personality, or background knowledge (Caravantes & Galán 2011; Feldman, Monteserin & Amandi 2015; Karampiperis et al. 2006). The importance of cognitive and learning styles is shifting from traditional learning to an online learning environment. Several models of cognitive and learning style in the literature have helped to improve the learning performance of the student (Caravantes & Galán 2011; Eberle, Schwarzinger & Stary 2011; Karampiperis et al. 2006). Research also confirms that personality and behaviour have an impact in processing patterns of information during learning. Thus, some researchers (Germanakos et al. 2008; Graf & Liu 2010; Kim, Lee & Ryu 2013) exploit theories from the psychological domain to model the learner. User interest and preference are also related to the style of displaying the information on the screen, which are relevant for the adaptation process.

The overview on learner model characteristics thus far shows that the research in adaptive learning acknowledges a number of user characteristics that are involved in learning activities. The next section will discuss the classification of these attributes from the perspective of some important relevant work from the literature.

2.2.2 Classification of Learner Model

Brusilovsky & Millán (2007) conducted a survey on user models in existing adaptive web-based education system. They identify the most popular attributes of user model in adaptive hypermedia as *user knowledge, interests, goals, background, individual traits,* and *context.* User (prior) knowledge has been identified as one of the most important learner features among those. Some adaptive educational systems only model user knowledge. The second most popular attribute in adaptive hypermedia is user interest. Goals represent the current learning objective and often change during the learning process, while background represents the experience outside the domain knowledge. These features, together with individual traits, are relatively stable and consistent during learning. The context of the work is usually used in ubiquitous learning, which includes physical, affective, and environmental dimensions.

In the review of (Granić & Nakić 2010), they proposed the initial user model by considering the human factor for the adaptive system. They suggested three broad categories of learner models:

1. *Personal user characteristics.* The attributes included in this category are general information (name, age, gender) and also individual traits (cognitive, personality, learning style).
2. *Previously acquired knowledge and skills.* This classification comprises prior experience, prior knowledge, and previously acquired psycho-motor skills of user characteristics.
3. *System-related data user characteristics.* Included in this category are the characteristics that are changeable and related to the particular systems, such as goals and requirements, preferences, interaction styles, and motivation.

A more comprehensive classification based on 20 taxonomies of the user model is reported by Medina-Medina, Molina-Ortiz & García-Cabrera (2011). They proposed an individual user model that characterises the adaptation of information in the system to specific features and needs. In their study, they categorised user models into five classifications:

1. *Personal data*. The features included in this category are name, surname, gender, address, age, and education/profession.
2. *Knowledge*. Included in this group are the visit numbers and knowledge of the subject.
3. *Experience*. This feature could be divided into two types, i.e. experience related to the subject and experience related to system navigation.
4. *Preferences*. This category consists of items and preferences for structuring the system.
5. *Interests.* This last category consists of knowledge subdomains, knowledge goals, and interesting concepts

Using a rigorous method of systematic literature review, Vandewaetere, Desmet & Clarebout (2011) searched for attributes that were used for learner models in the adaptive learning environment. They suggested three classifications of theoretical and empirical research for learner model, which consisted of:

1. *Cognition*. This category consists of the cognition-related characteristics such as working memory capacity, learning goals, prior knowledge, intelligence, cognitive style, and learning style.
2. *Affect*. This second category includes affective characteristics, such as frustration, confusion and delight, certainty and frustration, mood and self-efficacy.
3. *Behaviour*. Included in the last category are cognitive and affective states; the need for learner control and feedback; grades and the results of exercises.

In a literature study, Chrysafiadi & Virvou (2013b) reviewed various student modelling approaches published from 2002 up to 2012. They discussed the employment of the methods and classified the student models into *'knowledge and skills, errors and misconceptions, learning styles and preferences, affective and cognitive factors, meta-cognitive factors.'*

Considering all of the classifications reported here, future investigation is still warranted to examine the in-depth analysis of existing learner model in AES. Within the known models, not a single model can be adopted and directly implemented without knowing the background and environment. Each model has a different context and scope. Therefore, the keeping track with recent studies by revisiting and updating the findings is clearly important in this field. The evidence from recent studies shows that learner models in AES change and evolve over time. The constant changes of AES have become a motivation for this research to promote the potential learner's attributes that independent and could be adapted for a generic adaptive system. In the next section, the techniques to implement the AES is described, followed by a discussion of the proposed concept.

2.3 Techniques of Adaptive E-learning System

There are many variants of techniques that had been used to provide the best adaptation of learning resources to learners, from personal characteristics and preferences to knowledge levels and context. It has been noted that each approach has its strengths and weaknesses. Several publications have discussed the advantages and weaknesses of the various adaptation techniques in the literature (Drachsler, Hummel & Koper 2008; Kardan, Aziz & Shahpasand 2015). Before discussing methods and techniques, we introduce the conceptual model behind the adaptive e-learning systems. And then, the rest of this section discusses the adaptation method, including the challenges and limitations for each approach.

2.3.1 Conceptual Model and Design

Several models of AESs exist in literature. Most of the models adopt the model and design from adaptive educational hypermedia and intelligent tutoring systems (De Bra, Aroyo & Cristea 2004; Karampiperis & Sampson 2005; Sottilare et al. 2013). According to this model, an AES is composed of three basic elements: domain model, adaptation model, and user model. The relationship between the components in an AES is shown in Figure 2.5. The information in these elements is stored in the storage layer and linked directly with the runtime layer. The runtime layer performs the actual adaptation and delivers the results to the user.

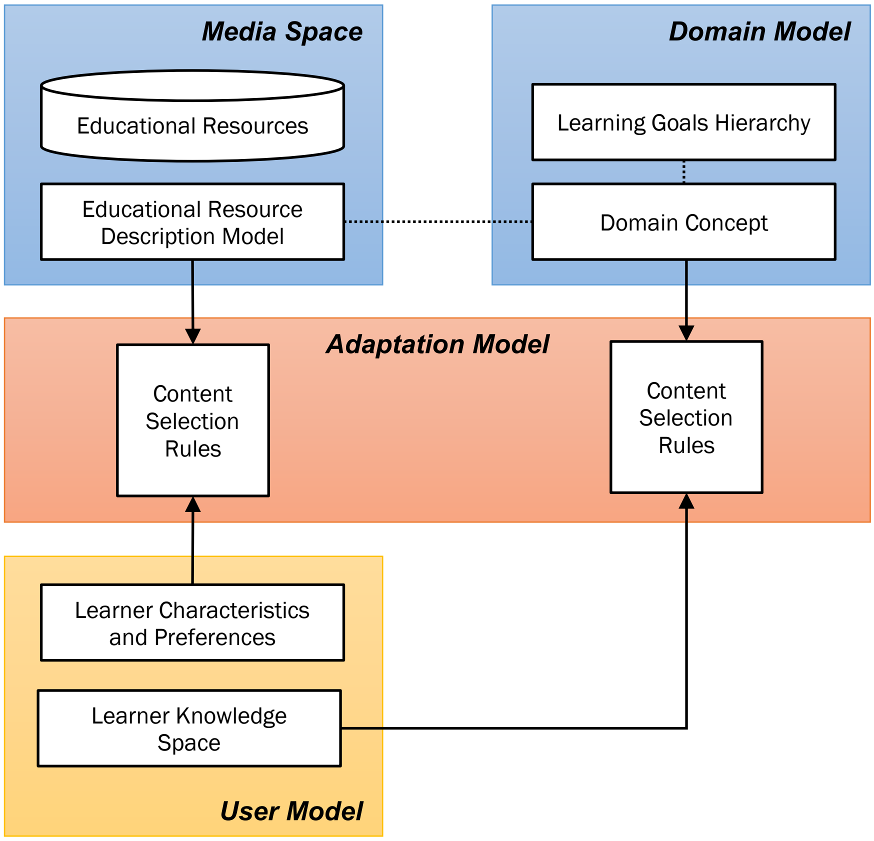


Figure 2.5 General elements and relation of AES (Karampiperis & Sampson 2005, p. 130)

A typical domain model consists of a domain concept and learning goals that describe the structure of a Subject in the real world. In adaptive learning, a domain model could be represented as a *concept hierarchy* or an *activity tree* that usually relies on a learning activity (IMS 2003). IMS describes learning activity as "*an instructional event or events embedded in a content resource*". In traditional learning, a learning activity is similar to a pedagogical unit, such as instruction, knowledge, assessment, etc. Each learning activity may have sub-activities or events nested in a deep level of another learning activity. It can also have a tracking status associated with each learner to provide individual learning experience. Figure 2.6 illustrates an activity tree or a domain model in an AES. Each node in this tree is a unit of activity. We can set the flow to navigate through the tree. For example, a learning activity should be started from lesson AA, then AAA, AAB, and lastly AAC. The idea of adaptive learning is to recommend a flow for each learner with their own choice according to their needs. Thus, the sequencing that will be delivered to each user is different from the other.

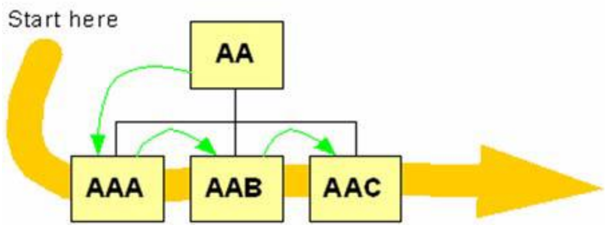


Figure 2.6 A Simple Domain Model (IMS 2003, p. 11)

The adaptation model (also known as adaptation model or pedagogical model) in AES serves as a bridge between the domain model and the user model. It has the responsibility of managing learning strategies, steps, and actions on what should a system deliver next (Sottilare et al. 2014). The adaptation model is often based on prerequisite relationships between concepts or learning activities. For instance, as illustrated in Figure 2.6, the concept AAA is a prerequisite for AAB. This means the students should understand AAA before they study AAB. The choice of strategy varies, by following the interest of the learner, taking an assessment, or requesting assistance. Regardless of the choice, an AES should be ready to serve the next activity for the user based on the instructional strategy. The strategies for adaptation model are usually grounded by the researchers' pedagogical theories, such as learning style, activity theory, cognitive style, learning traits, or affective state (Caro et al. 2014; Hernández, Sucar & Arroyo-Figueroa 2013; Özyurt & Özyurt 2015; Peña-Ayala, Sossa & Méndez 2014).

The last component of AESs is the user model (also known as learner model), which always becomes a central issue for AES research. Although there are many successful stories of AES implementation to learners, the fact is that every study has used a different user model to represent individuals. As there is no default standard to represent a user model, there are several investigations that seem to shape the characteristics of user modelling in AESs. Understanding the complexity of user modelling is important if we want to renew the body of knowledge on user modelling in adaptive e-learning systems. The topics of variables and properties of user modelling will be explored separately in Chapter 4. In the remainder of this section, we present three methods that are commonly used for adaptive learning.

2.3.2 Rule-based Adaptive Models

The rule-based technique is still becoming an effective way of handling knowledge in design and implementation of adaptive e-learning systems. Rule-based technique is derived from simple conditional structures of IF/THEN/ELSE statements (García et al. 2009). It can also provide feedback for each learning activity taken by user. Generally, the implementation of rule-based technique in e-learning domain is by employing the adaptation rules to select and sequence the learning resources. The rule-based model also could be combined with the decision tree, with a tree as pre-prescribed content material. A set of rules is used in a rule-based model to determine the conditions that change the learning path for learners. Although the rules seem relatively simple, their implementation in real environments is a difficult and tedious task. The problems of inconsistency and insufficiency arise as the number of rules increase (Karampiperis & Sampson 2005). For many years, numerous studies have attempted to solve this problem by utilising various methods. Regardless of the problems, the rule-based model is still a popular option for adaptive e-learning.

In (Graf et al. 2009), the authors investigated how the students interact with a learning management system and then, using a rule-based approach, suggested learning activities based on students' learning styles. Similarly, Popescu (2009) analysed the student behaviour through learning style, media type, grade, time spent on a test, and interaction style. They then proposed a rule-based approach to recommend the learning object in an AI subject. Benlamri & Zhang (2014) combined an ontology space and a rule-based reasoning as an adaptation mechanism of learning sequence and learning content, based on learner's activity, environment, and background. (García et al. 2009) proposed a rule-based method for mining data in e-learning and web-based adaptive educational systems. They used the form of IF-THEN rules to discover insightful information related to student's behaviour and then recommended the results to improve teaching contents.

2.3.3 Principle-based Adaptive Models

Principle-based technique refers to the practical approach, which is derived from different kinds of algorithms, such as machine learning, heuristics, evolutionary algorithms, recommender techniques, or data mining. There are numerous researches in this classification that have extensively been utilised in adaptive e-learning systems. For instance, (Su et al. 2011) proposed a Personalized Learning Content Adaptation Mechanism that implemented data mining techniques (clustering and decision trees) to deliver personalised learning content to learners. Previous studies have also shown a significant increase in utilising an evolutionary algorithm for optimising the sequence of learning paths by employing genetic algorithm, particle swarm optimisation, and memetic algorithm (Acampora, Gaeta & Loia 2011; Chen 2008; de-Marcos et al. 2011).

The review of studies in data mining has been reported in (Romero & Ventura 2010). The most popular adaptation based on these techniques was intended for providing feedback, predicting performance, content recommendation, and analysing the learning behaviour. Several data mining approaches for clustering, classification, and association analysis have been widely used for mining student's data. For example, fuzzy clustering statistic correlation, grey relational analysis, K-means clustering, and fuzzy-association-rule have been applied to support adaptation in several e-learning platforms (García et al. 2009; Jun-Ming et al. 2006; Salehi, Nakhai Kamalabadi & Ghaznavi Ghoushchi 2014).

2.3.4 Hybrid Adaptive Models

The hybrid technique combines some algorithms or approaches to deliver better result in recommendations. For example, Salehi, Nakhai Kamalabadi & Ghaznavi Ghoushchi (2014) use the combination of sequential pattern mining and collaborative filtering to provide recommendation of learning material. Similarly, (Chen et al. 2014) utilise a hybrid recommender system using collaborative filtering for content discovery, and then apply sequential pattern mining to filter items according to common learning sequences. In (Klašnja-Milićević et al. 2011), the authors propose Protus that automatically adapt to the interests and knowledge levels of learners. This system processes the clustering-based on learning styles and analyse the learners' habits through AprioriAll algorithm. The combination of collaborative filtering and vector space model also improve the recommendation quality in e-learning, as has been demonstrated in (Wang & Huang 2011).

Based on these reviews, the techniques proposed in adaptive learning has varied and evolved over time. The classic rule-based approaches have been widely acknowledged and accepted as the general approach for adaptive learning. However, the execution will rely on a complete taxonomy of information related to the student and the educational domain. On the contrary, the principle-based approaches were proposed mostly to handle the lacking of the information about the domain. The ideas for the adaptation was expanded from rule-based matching to classification and optimisation models. The limitation of this approach is that the models often built for general purpose. The application will need many customisations in practice. Furthermore, the hybrid method is intended to provide a solution by combining several methods and knowledge altogether. This approach commonly used for solving a significant problem with many variables to be considered. Thus, the implementation will often need more considerable resources. The AES in this thesis will be developed utilising a rule-based system. The rules were stored in the form of a knowledge base. In the next section, the principal finding of the current investigation of a knowledge-based system was presented.

2.4 Knowledge-Based System (KBS)

Studies of knowledge base systems are well documented. It also well acknowledged that the term "*knowledge-based system*" suggests that it deals with a system that is based on the use of knowledge. In particular, a KBS can be defined as a computer system that uses knowledge about some domain in order to deliver a solution concerning a problem (Ammar-Khodja, Perry & Bernard 2008). This definition confirms that the solution from the system should be the same as the solution from a domain expert. The discussion in the following sections presents the historical background and evolution of KBSs, followed by the structure, development lifecycle, and knowledge acquisition process.

2.4.1 History

The early generation of KBS was an expert system. In an expert system, the domain knowledge is encoded explicitly into a knowledge base (KB) and it uses automated inference to support the expert. These two components, knowledge base and inference engine, are basic composition of the expert system (Figure 2.5). The knowledge base is usually composed of a set of facts and rules that express domain-specific knowledge, while the inference engine has a role in controlling the order of the rule activation. The characteristic of an expert system is that it applies a specific domain of knowledge to solve a specific problem of data in order to produce problem-specific conclusions.

For many years, expert systems have proven useful in many real-world applications and can be found in a wide variety of domains, like medicine, chemistry, agriculture, and science. Some well-known and documented expert systems in literature are: (1) MYCIN, which could identify infectious bacteria in blood and urine samples, (2) CADUCEUS, a system that focus on diagnosis of diseases, (3) PUFF, an interpretation of respiratory tests for diagnosis of pulmonary disorders, (4) QMR, that could assist physician diagnoses of over 4000 disease manifestations, (5) DENDRAL for identifying molecular structure of new compounds, (6) CRYSALIS to capture electron density maps in protein crystallography, (7) MOLGEN, the tool for manipulating DNA, (8) PROSPECTOR that guided geologists in the exploration of ore deposits, and lastly, (9) R1, the expert system to configure VAX (Buchanan & Shortliffe 1984; Feigenbaum, Buchanan & Lederberg 1970; Gevarter 1984; Lundsgaarde 1987; McDermott 1982).

The second generation of KBSs was inspired by the concept of heuristic classification that uses abstraction to perform inference (Mitchell 1997). This idea was a push to a more sophisticated reasoning mechanism that might separate knowledge at the symbol level and knowledge level. The symbol level in an expert system is used for encoding a KB and inference engine, while the knowledge level is used to express the abstraction. This philosophy inspired numerous frameworks to develop the problem-solving methods for reasoning process in a KBS. This problem-solving method has been used in many modern expert systems.

2.4.2 Structure

As previously stated, the solution from the KBS should be the same as the solution from the domain expert. To achieve this purpose, a KBS is designed to capture and then apply a specific knowledge from a domain expert. Thus, the process of building a KBS usually involve both. a knowledge engineer and a domain expert (Russell & Norvig 2016). A knowledge engineer typically investigates the important concept from the domain expert and then determines the formal representation of the objects and relation in the domain. This transfer of knowledge and transformation of problem-solving from domain expert is acknowledged as Knowledge Acquisition. Figure 2.7 illustrates the general structure of a KBS and relationship between knowledge engineer and domain expert.

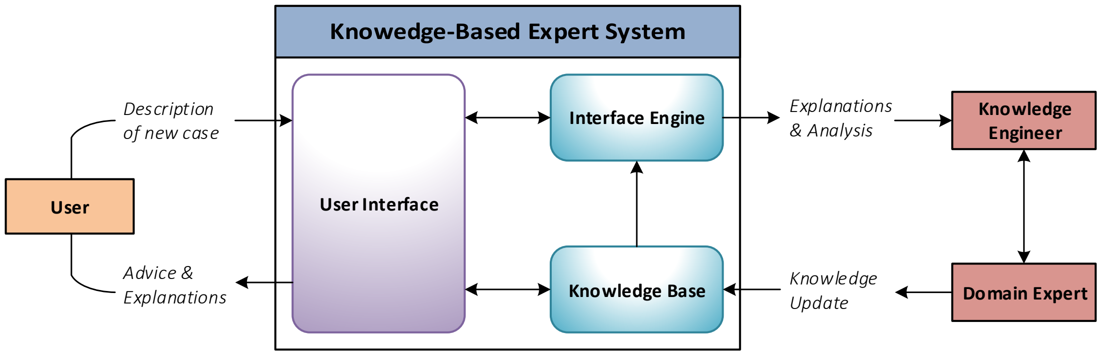


Figure 2.7 Interaction between knowledge engineer and domain expert with KBS (Buchanan & Shortliffe 1984, p. 7)

In KBSs, all the cumulated facts and important concepts that already known is stored in a repository as a knowledge base. The knowledge base is not a database that stored all information in a structured manner. It contains concepts, rules, and experiences of the way experts have solved problems in previous actions. The information in a KB is important for understanding, formulating, and solving the problems. A key role in a KB is the mechanism in representing the knowledge as the knowledge from expert should be represented formally. Knowledge representation emphasises in structuring and manipulating information from the knowledge acquisition process.

After all knowledge has been orderly stored in the knowledge base, a reasoning mechanism using a logical technique is used to operate the knowledge base. A KBS performs reasoning using an inference engine. The inference engine plays an important role to regain and determine how to use the knowledge and present it to the end user. In contrast to a knowledge base that plays a passive role by storing the knowledge, an inference engine plays an active role in a KBS by interpreting and structuring the rules. This component is the brain of the KBS, which handles the communication between the knowledge base and end user interface. In fact, it can be said that these two features: knowledge base and inference engine, are the components that distinguish KBSs from other systems. The expert should carefully select the methods for representing knowledge in a KB. The things that should be considered are, the methods to handle the enormous amount of encoded knowledge, and the reasoning mechanism on the supplied knowledge. Central to the most reasoning systems is the use of logic. The most applicable and widely used logic for reasoning in KBSs is propositional logic and first-order logic (Russell & Norvig 2016).

Among a variety of ways to represent a knowledge base, a rule-based system was the most popular in the early generation. Rule-based-systems use rules as their basic inferencing mechanism. The basic structure of a rule-based system is a set of IF-THEN statements. The mapping statement is known to be true (as condition) to a conclusion (result). This mapping is performed using logic, like in proportional logic. For example, IF "*temperature is hot*" THEN "*turn on the AC*". Many studies in the early development of KBSs utilised a rule-based system for pathology and medical classification (Buchanan & Shortliffe 1984; Lundsgaarde 1987).

The logic approaches for inference engine mechanism consist of forward chaining and backward chaining. As its name implies, forward chaining works from the case to the conclusion. When a case is presented to the system, the knowledge base searches for a rule that satisfies the condition. This process is repeated until user gets the expected result. On the other hand, backward chaining works from the conclusion to the case. The desired result is used as a starting point, and then the knowledge base searches for the rules contain the required statements in their results. The process is repeated until the condition is met. The inference terminates when all rules are verified against the case. The forward chaining method was designed to work if all the facts are available, while the backward chaining method was designed for a domain where no facts might be available. It works backwards by asking a question and gradually refining the result. Both approaches might also work together as a hybrid forward and backward chaining system.

2.4.3 KBS Development Lifecycle

As for conventional software development, a KBS also has a specific lifecycle model. The lifecycle is a framework that provides a guideline to be followed for knowledge-based system development. It has a structure and a detailed plan describing the tasks or steps at each phase. The lifecycle serves as a methodology for improving the quality and overall KBS development process. The early KBS lifecycle was Generic Tasks as introduced by (Chandrasekaran 1986). Generic tasks use building blocks for KBS development. The building block is basically a hierarchical classification that provides information about (1) a task specification and form of knowledge for input and output; (2) a task related to domain knowledge and specific organisation; and (3) a control regime for the task. The hierarchical classification was demonstrated for MDX project to define a disease hierarchy for identifying a patient's case. Each of the concepts in the classification hierarchy contains 'how-to' knowledge for the diagnostic task.

A different approach to construct a knowledge-based system was *"the knowledge-based system development life cycle (KBSDLC)",* as presented by (Weitzel & Kerschberg 1989). KBSDLC uses a sequential progression, which consists of five phases, i.e. identification, conceptualisation, formalisation, implementation, validation, conversion and operation. The model was straight and simple, but helpful to simplify the development process. KBSDLC has been implemented in the Medclaim project for rapid prototyping. The prototype was validated through a parallel test of expert reviews. As the prototype of the KBS was ready, problem understanding, system design and development were evolved together.

In 1990, (Agarwal & Tanniru 1990) proposed the expert systems development life cycle (ESDLC) which was derived from the traditional system development lifecycle (SDLC). The ESDLC consists of four phases: (1) problem identification; (2) system development; (3) transfer to production; and (4) operation. The only phase that differentiates between ESDLC and SDLC is phase 2, where in ESDLC it includes conceptualisation, formalisation, implementation, and testing of the SDLC.

The comprehensive lifecycle model for KBSs was knowledge acquisition and design support (KADS) (Wielinga, Schreiber & Breuker 1992). KADS uses a modelling activity where every layer to develop a knowledge-based system is described in a model. KADS has a four-layer framework, namely, domain knowledge, inference knowledge, task knowledge, and strategic knowledge. The domain knowledge is the first layer that conceptualises the domain theory of a particular application. It is denoted by concepts, properties, two types of relations and structures. Inference knowledge contains the source of knowledge that operates on a meta-class. Task knowledge contains knowledge which combines the inference to achieve a particular goal. The elements included in task knowledge are task, control terms, and task structure. Strategic knowledge determines the important goals for solving a certain problem.

An improved version of KADS is known as CommonKADS (Schreiber et al. 1994). The KBS in CommonKADS is built by modelling the organisation and application context. CommonKADS provides four models to construct a KBS: organisation, tasks, agent, and communication model. The organisation model analyses the problems and opportunities for KBS development. The tasks model describes the task which will be performed in the organisation. It is represented as a hierarchy. An agent acts as an actor that executes a task. It can be a human, software, or other entity that can execute a task. The communication model is responsible for modelling the communication between agents. The divergence between CommonKADS and KADS is that the CommonKADS model provides more details on the conceptual structure of knowledge and abstraction through knowledge roles and ontologies. Many projects, both commercial and academic, have supported the CommonKADS model as a reliable knowledge engineering methodology for constructing a KBS.

2.4.4 Knowledge Acquisition (KA)

Knowledge acquisition is the process of extracting, structuring, and populating a knowledge base. The process is usually performed by a human expert or automated process based on available data. Building a KBS will require knowledge that is captured from human experts and translated into a format that suitable for the KBS. Traditionally, knowledge is obtained from a domain expert by a knowledge engineer. The first task of a knowledge engineer is to extract knowledge from a domain expert. This could be carried out using various methods, such as interviews, questionnaires, collecting cases, observation and protocol analysis. However, this task is never easy as it raises a number of problems, like inaccuracy and incomplete knowledge.

If the knowledge engineers can find the problems early, they have to arrange a follow up interview. However, the cost is expensive if they discover the problem later as they have to rebuild the system. This issue has received considerable critical attention. An effort to improve the acquiring of expert knowledge was introduced by using knowledge elicitation approach. During the process, all knowledge and problem-solving strategies are identified. The process starts from a wide scope in the early stage and becomes narrower focused on the specific knowledge.

After the information has been collected from the expert; it needs to be identified into usable knowledge. As the information in the early stage is quite general, analysis is needed to uncover the knowledge. The commonly used framework in structuring the knowledge is Knowledge Analysis and Design Support (KADS) (Schreiber et al. 2000). Knowledge acquisition advanced further with the use of more automated techniques. A well-known system for aiding the construction of such a tool is Protege. This tool allows the knowledge engineer to tailor generic tasks to the domain to be modelled. It is however, quite conceivable that incremental acquisition can make a significant contribution to knowledge acquisition. Ripple-Down Rules is a form of incremental KA approach, which is discussed in the next Section. The implementation of this method is discussed in Chapter 5.

2.5 Ripple-Down Rules (RDR)

This section presents a review on Ripple-Down Rules as a knowledge-based technique to acquire knowledge from an expert. The first section describes the background and philosophy of RDR. Then, several variations of RDR approaches are described in the next section. Despite the variations, the main ideas embodied in each implementation are: KBSs that are grounded in cases, an exception structure and a simple knowledge acquisition technique performed by the expert and which encourages incremental development, maintenance, and validation of the knowledge base.

2.5.1 Background and Philosophy

Ripple-Down Rules (RDR) is a knowledge acquisition (KA) technique proposed by Compton et al. (Compton et al. 1991; Compton et al. 1992), which grew out of long term experience of maintaining a medical expert system. The two major issues in early RDR research concerned with (1) acquiring knowledge based on the context, and (2) maintaining a large amount of knowledge in an expert system (Compton et al. 1992; Richards & Compton 1997). The first problem was raised because the most successful applications of expert systems need to interpret data correctly according to their context. For example, the experts tend to justify easily a particular recommendation for a patient, but sometimes they give a different treatment for another patient. This happens as the result of the captured knowledge which is associated with a different context. Knowledge is something that is clearly stated and remain true over time. RDR uses knowledge in the context which it is acquired to ensure that only valid rules defined by experts can be entered into a knowledge base.

With its underlying philosophy, RDR distinguishes it from other expert system techniques. In the case of RDR, the rules are made to be corrected. Having given an overview of RDR, it is worthwhile to consider why it was chosen as the knowledge acquisition and representation method on which to base further development. Four major reasons for using RDR include (Richards 2009):

1) RDR has addressed several problems that have always been associated with KBSs, which are the knowledge acquisition, maintenance and validation problems.
2) RDR offers a direct involvement system that gives the user control and ownership of the system.
3) RDR provides knowledge in context, which was identified as important in allowing knowledge reuse.
4) RDR provides the availability of the existing software and accessibility to those most familiar and knowledgeable of RDR.

2.5.2 Approaches and Variations

Several variations of RDR are found in the literature. The following subsections introduce basic Single Classification, Multiple Classification, and a summary of the key variants of RDR.

1. Single Classification RDR (SCRDR)

At the beginning of the RDR development, it was originally meant to handle single classification tasks (Compton et al. 1991; Compton et al. 1992). The form of a Single Classification RDR is basically a binary tree with a root node named default node. The default node is a node that is always true and gives a default answer to every single query. Each node in RDR is associated with a rule, which has a condition, conclusion, and cornerstone case. Each node in the tree also has two types of edges, namely, *except* and *if-not*. By using algebraic foundations (Scheffer 1996), a SCRDR can be formally notated as the set of <*rule, T, F, cc*> where *T* is TRUE branch, *F* is FALSE branch, and *cc* is the associated cornerstone case. More clearly, an SCRDR is depicted graphically as in Figure 2.8.

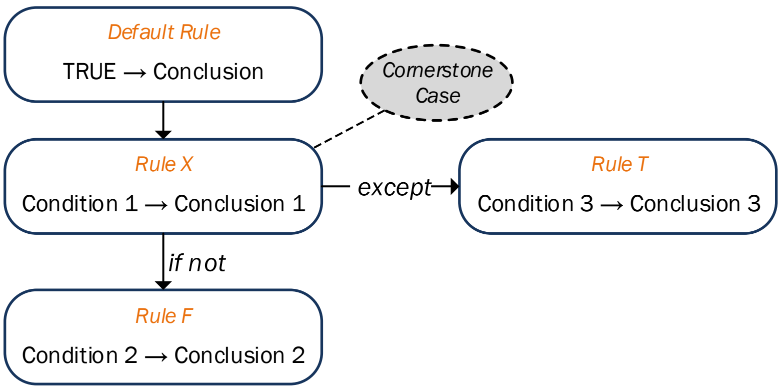


Figure 2.8 An Example of SCRDR

As shown above, rule X consists of condition 1 and conclusion 1. During the acquisition process, a new case with condition 2 is presented to rule X. If it is concluded that condition 2 does not match condition 1, then a new rule F is added in the false branch as a child for rule X. Otherwise, when a case 3 is presented and the expert concludes that condition 3 matches with rule X, but conclusion 1 is invalid or misclassified, then a new rule T is created. Rule T is a true branch and located at the exceptional structure to provide rule pathways. Furthermore, every case that caused the misclassification is needed to be stored in association with the new rule. This case is referred to as the *cornerstone case*. In fact, the cornerstone case has become the most important feature for knowledge maintenance in a knowledge-based system (Compton

& Jansen 1990). As RDR uses a failure-driven approach to acquire knowledge, the existence of a cornerstone case is required to maintain changes in the knowledge base (Richards 2009).

The knowledge acquisition process below illustrates the operation of an SCRDR system more comprehensively:

1) An empty SCRDR knowledge base starts with one rule that classifies all cases as a default classification.
2) Then, the case is evaluated against the top rule.
3) For each node, when it satisfies the rule, then evaluate the true path (except) and ignore the false path (if not).
4) On the contrary, when it does not satisfy the rule, then follow the false path and ignore the true path.
5) Lastly, the true node taken through the pathway gives the conclusion for the presenting case.

Figure 2.9 depicts an example of a Single Classification RDR knowledge base that classifies objects into two classes: *fly* and *not fly*.

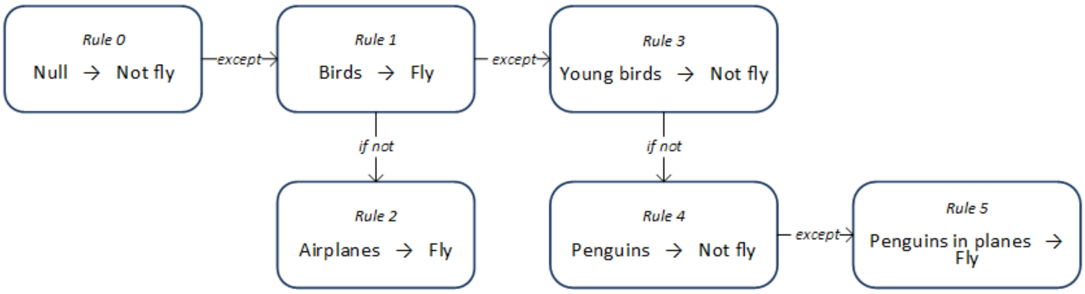


Figure 2.9 An Example of SCRDR Knowledge Base

In the above picture, each rule can be seen as a pathway that leads from itself back to the root (Rule 0). The sample SCRDR knowledge base shown in Figure 2.9 above can be formed in the following manner:

- At first, the only node in the knowledge base tree was the root node or rule 0. This root rule has a default value that is always true and a null with default conclusion. All cases presented to the tree initially satisfy rule 0.

- Case 1 is presented to the SCRDR knowledge base and the expert disagrees with the conclusion from rule 0. The expert creates a new rule, namely rule 1, as an exception of rule 0. They also have to differentiate the feature between rule 1 and rule 0. Case 1 then becomes the cornerstone case for rule 1 since it causes rule 1 to be created.
- Case 2 is presented to the SCRDR knowledge base and evaluated against the tree. The last true path for the case is rule 1. However, case 2 evaluates to false against rule 1. The new rule is created as the false path from rule 1. Case 1 becomes the cornerstone case for rule 1.
- Case 3 is presented to the SCRDR knowledge base and evaluated against the tree. The last true path for the case is rule 1, but the expert is dissatisfied with the conclusion. So, the expert creates a new rule 3 and case 3 becomes the cornerstone case for rule 3.
- Case 4 is presented to the SCRDR knowledge base and evaluated against the tree. The last true path for the case is rule 3. However, the presented case evaluates to false against the condition of rule 3. The new rule is created as the false path from rule 4. Case 4 became the cornerstone case for rule 4.
- Case 5 is presented to the SCRDR knowledge base and evaluated against the tree. The last true path for the case is rule 4. However, the expert is dissatisfied with the conclusion and offers a new conclusion, created as a rule 5. Case 5 becomes the cornerstone case for rule 5.

To summarise the above example, the RDR ensure to validate a KB by maintaining every piece of information during knowledge acquisition process. The presented case is stored alongside the rules and associated as the cornerstone case. If the expert adds a new rule, then the system offers some conditions for the new rule. This mechanism ensures that the cornerstone case of the parent does not fire on the new rule and gives the current case the correct conclusion. The depth and breadth of the SCRDR knowledge base depends on the degree of overlapping classification and the order of the presenting cases (Cao & Compton 2005; Scheffer 1996).

2. Multiple Classification RDR (MCRDR)

In 1995, Kang et al. introduced an extended version RDR called Multiple Classification RDR to address compound classifications problems in SCDRD (Kang, Compton & Preston 1995; Kang, Gambetta & Compton 1996). MCRDR supports the philosophy and strategies that made RDR valuable, like alteration of the underlying binary tree structure and inferencing process. However, it expands the capability with multiple classifications. Thus, it deals with the task that requires multiple independent classifications. Every rule in MCRDR only has exception nodes. If the presented case is satisfied with a rule, then it will be evaluated against all child nodes. The process is repeated until all the child nodes are satisfied by the case or no more nodes are left to be evaluated. Then the conclusions are given from the last satisfied rule in each path.

Figure 2.10 illustrates an example of MCRDR KB with each node consisting of a set of conditions, classification, and references to all child nodes. For example, a case with condition x = (1, 4, 6) will fire with the multiple classifications Class 1, Class4, and Class 6. Class 4 will be reached from node 3, while Class 6 will be reached from node 5.

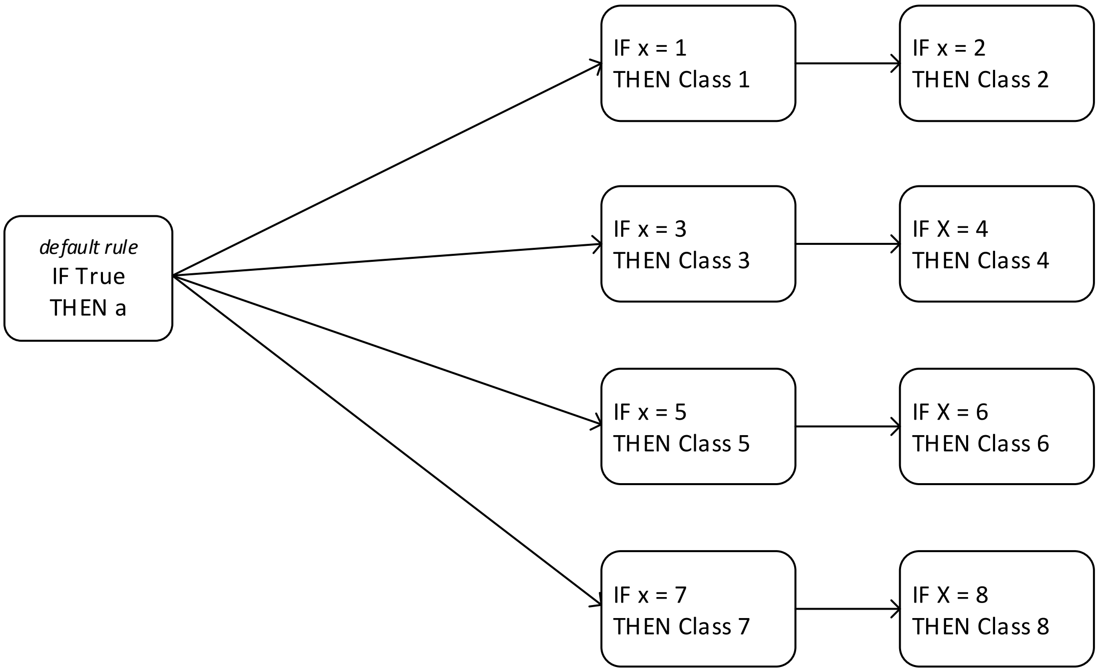


Figure 2.10 Example of Multiple Classification RDR KB

The knowledge acquisition process in MCRDR is similar to SCRDR, but with consideration of the case that requires more than one classification. Below is illustrated the KA of MCRDR system:

1) The system provides MCRDR knowledge base with a list of classifications.

2) Then, the case is evaluated against the top rule.

3) For each node,

    a. When it misses the classification, then expert creates a new rule to add the missing classification.

    b. When it does not satisfy the rule, then expert creates an exceptional rule to fix the incorrect classification.

4) If the case is given a true conclusion, then return to step 1 with a new case.

Through the above process, it is possible for the expert to incrementally build and improve the knowledge base. This process is simple and natural by gradually examining the case to ensure that system has classified them correctly. However, there are three requirements that should be considered when creating a new rule in MCRDR. First, the expert must ensure to state the classification to the system. This is a trivial process, but if they do not desire to make a classification, it may cause unwanted situations during the validation phase (Kang, Compton & Preston 1995). In the worst case, the size of the KB will inflate and be redundant if the expert expresses the same thing in different classifications. So, the experts must ensure that after created a node, they should use it for the future child.

The second requirement is location of the rule. This is crucial, for instance the new rule could be added to the root node if the missing classification is found. Moreover, if the case is classified wrongly, the new rule will be added as an exception to the nodes that prove the wrong classification. Finally, the acquiring of a rule condition is also important. When the expert states the condition that they believe will imply the particular classification, they usually do not provide a specific rule. This might cause conflicts with past seen cases. Thus, additional validation will be required and might require further refinement of the rule.

3. Nested RDR (NRDR)

Introduced by (Beydoun & Hoffmann 1997b), Nested RDR presents RDR a new way to model domain knowledge at the knowledge level. NRDR is built based on SCRDR but consists of any number of SCRDR trees. The representation of a concept defined by the experts has also distinguished NRDR from SCRDR. Thus, the concept can be defined from the KA process and in turn can be used as an attribute in rules for another RDR tree. This mechanism will result in a hierarchy of RDR trees. The main advantage of NRDR is the ease of using knowledge reusability. NRDR addresses the issue of knowledge repetition in RDR by providing the nested concepts that can be re-used in many places of the KB. Different from other KBs, concepts in NRDR can be present during KA process. (Beydoun & Hoffmann 2000a) noted that NRDR has higher modularity than MCRDR by enforcing a hierarchy of RDR trees which isolates interactions between different parts of the KB. They also reported that NRDR has a more compact KB, as the number of inference paths is lower than SCRDR or MCRDR, but can cover more cases.

4. Other Variations

Besides NRDR, some variations of RDR exist. Most of them are an extension of MCRDR rather than RDR. For instance, Rated/Weighted Multiple Classification RDR by (Dazeley & Kang 2003a; Dazeley & Kang 2003b); MCRDR with formal concept analysis as introduced by (Kim & Compton 2004; Richards & Compton 1997; Richards 1998); and also Bayesian Threshold with MCRDR proposed by (Cho & Richards 2004). However, we do not cover these in detail here.

2.5.3 Strengths

A significant strength of RDR is that knowledge acquisition and maintenance are easily achieved. The simultaneous incremental KA with RDR as the underlying knowledge representation could ease the KB development process. The KB grows incrementally based on the cases seen by it. A new rule is only added when the KB gives a wrong conclusion. RDR also organises the structure very well without messing up the knowledge in the KB. When dealing with a new case that is never seen before, the expert only needs to identify the features that differentiate the new case with the rest of the cases that are already stored in the KB. In terms of building the KB, RDR does not allow the expert to add any rules as they would give a different conclusion. Even though the knowledge seems incomplete, the hierarchical incremental KA process that captures the cases one at a time makes the KA task more effective. This mechanism would keep the system valid as it always consistent with the existing knowledge (Beydoun & Hoffmann 2001; Compton et al. 1991).

The other advantage of the RDR is that this form of knowledge acquisition results in compact and efficient knowledge bases. It might be expected that incremental addition of knowledge where knowledge was only added (as a refinement) but never changed, may result in very large knowledge bases with much of the knowledge being repeated knowledge. However, simulation studies show that the size of the knowledge bases are comparable to those produced by machine learning (Kang, Compton & Preston 1995), and that there is a significant increase in size only when a random choice is used by the expert. In studies on a human developed MCRDR knowledge base (~3000) rules, only 10% compression could be achieved (Suryanto & Compton 2000)

2.5.4 Limitations

Despite great success in a wide range of application areas, the current RDR-based systems have been criticised for their limitations in supplying an explicit model of the domain knowledge (Beydoun & Hoffmann 1997b; Compton et al. 1991; Kang, Compton & Preston 1995). This means that the RDR methodology does not support higher-level models, especially abstraction hierarchies. RDR assumes a simple attribute value representation of the world and supports only rules rather than inheritance or other deductive reasoning from an ontology.

To address this lack, some work has already been done on evolving hierarchies in parallel with RDR (ROCH; Martinez-Bejar, Benjamins et al. 1998), discovering abstraction hierarchies from MCRDR using formal concept analysis (Richards & Compton 1997), and modelling domain knowledge with simultaneous knowledge acquisition using NRDR (Beydoun & Hoffmann 1998). Another limitation of RDR is the repetition within the knowledge base (Richards & Malik 2002). However, the repetition problem is not a serious impediment of RDR as addressed in (Suryanto & Compton 2000).

Over the past two decades, there have been numerous studies of RDR. However, most studies have only focused on the implementation of the new problem domain with a similar step when dealing with acquired knowledge. The new cases will be presented to an expert who decides the conclusion. The features of visualizing, debugging and evaluation were introduced with some issues under consideration {Richards, 2009 #1350}. In this study, a statistical module was developed to compute the knowledge acquisition process and visualize the efficiency of the process. We used various functions to measure the readiness of the knowledge base. This technique allows an expert to determine the continuation of the KBS development project.

2.5.5 Applications

The Ripple-Down Rule has been a variety of application, in both research and commercial field. These include chemical pathology interpretation (Preston, Edwards & Compton 1994), ion chromatography method in analytical chemistry (Ramadan et al. 1998), web document classification (Kim & Compton 2004), email management assistant (Ho, Wobcke & Compton 2003), object recognition in RoboCup (Shirazi & Sammut 2008), CAD design assistant (Hamade et al. 2010) question and answering system (Nguyen, Nguyen & Pham 2014), and schema mapping (Anam et al. 2015). Richards (2009) reviewed the perfect history of RDR research over the past two decades and its applications. This section highlights some types of RDR applications.

The majority of RDR applications have been developed to address various classification problems. The experience with GARVAN-E1 as a medical expert system has led to the first RDR system known as Pathology Expert Interpretative Reporting System (PIERS), which was incrementally built using RDR (Compton et al. 1992; Preston, Edwards & Compton 1994). In the four years, over 2000 rules were added to the knowledge base during routine use, allowing the knowledge base to automatically interpret chemical pathology results and generate reports. Following the success of PIERS, Labwizard, which was developed by Pacific Knowledge System, uses a variant of MCRDR to provide interpretation for medical chemical pathology reports. In their study, (Compton et al. 2006) reported that their system could process up to 14,000 patient reports per day through 23 RDR knowledge bases with about 16,000 rules built over 29 months.

Another application of RDR is the development of an e-mail management assistant that handles the process of e-mail management (Ho, Wobcke & Compton 2003). By utilising RDR, the system could achieve very high levels of accuracy when dealing with the classification of messages. The accuracy level in classifying messages is reported between 95%-99% over the experimental period. (Kim & Compton 2004) introduced an RDR-based document management and retrieval system. The document is annotated by users and then browsed using the concept lattice of formal concept analysis. They reported that the combination of RDR to represent knowledge in document management and concept lattice is a powerful tool for supporting the open management of documents. Prayote & Compton (2006) have demonstrated an RDR-based approach to network traffic intrusion systems. They use RDR to randomly segment the problem space in network intrusion into sub-spaces of homogeneous traffic. The finding reports that the system successfully detected traffic anomalies, with low false positive and false negative rate.

All of these systems have used RDR for the classification task. Another popular application is for the searching task. Search based problems are characterised by the requirements of the knowledge-based system to solve complex combinatorial problems. Here the RDR knowledge base is employed to guide the search within a constraint satisfaction framework. Significant analyses and discussions on the searching task were presented by Beydoun & Hoffmann (2000a). In their previous study, Beydoun & Hoffmann (1997b) proposed Nested-RDR and employed it to build a chess-playing system. Chess players incrementally define and refine a hierarchy of concepts for the chess domain, and the suggestion of a chess move is defined on these concepts. With a similar approach, Bekmann & Hoffmann (2005) used Nested-RDR to guide a genetic algorithm-based search to discover VLSI chip layouts. Genetic algorithms rely on a fitness function to evaluate the quality of the variations produced by mutation at each generation. Nested-RDR was used to construct a knowledge base to heuristically define the fitness function.

In (Richards & Compton 1999), MCRDR is used to address the NP-hard problem of room allocation in a system called SISYPHUS-I. In their system, MCRDR suggests a possible allocation of a person to a room. If the user does not agree with the recommendation, rules are added using features defined on the person and the room. By implementing this approach, RDR does not need substantial modelling before a solution can be found. In contrast, RDR uses the data provided in the documentation to build a knowledge base with minimal a priori analysis. Then, the formal concept analysis is used to automatically generate an abstraction hierarchy. This approach was proven sufficient to map people to rooms and handle allocating and tracking resources.

Several researchers have also applied RDR to solve complex control problems. Here the conclusion or output of the RDR knowledge base directs an action. Shirazi & Sammut (2008) used RDRs to learn control of the actions in flying a plane. Here RDRs were employed to acquire tacit knowledge that experts subconsciously employ in flying a plane. Bekmann & Hoffmann (2005) also used Nested-RDR to guide the mutations within a Genetic algorithm-based search for VLSI chip layout. With only 30 hours of training, their HeurEAKA system produced chip layouts comparable to those produced by domain experts using industry standard algorithms, developed over years.

Finlayson & Compton (2004) applied the RDR framework to the development of multi-agent teams in a complex real-time environment of Robocup 2-D soccer simulation. Here the concepts of RDR were used to acquire the knowledge to guide individual agents within a cooperative setting. Although these approaches have acquired control knowledge to guide specific tasks in isolation, the vision system often requires multiple components to operate in concert with each other and may potentially interfere with each other. Learning of control knowledge in vision domains also requires learning at multiple levels.

This thesis presents a new mechanism to broaden the applications of RDR. To date, there has been limited investigation of the use of RDR for the educational context. Considering there is an increasing need for e-learning solutions to include intelligent methods for supporting learning activities. This thesis provides evidence that the RDR could be supported to facilitate the adaptation and personalisation of e-learning system. The use of RDR as the core engine for adaptation has allowed experts to store their knowledge for a learning domain.

2.6 KBS Evaluation Methodologies

Evaluation in software development is often introduced as validation and verification (V&V) activities. Although validation and verification are often linked together, the meaning of both terms are totally different. The term validation is commonly referred to as "*the process of building the right system*" for users. According to IEEE (IEEE 2005), validation is "*the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements*". The definition implies that validation is required to ensure that the system has met the criteria set by the users. This process is different from verification, which evaluates the product according to the design and specification requirements. It could be possible that a system has passed the verification test but has failed the validation test. Many systems might be rejected by users if the validation result is not satisfactory. Therefore, the validation activity is intended to demonstrate that the system can reach the correct conclusion with respect to the user's needs and requirements (Vermesan & Coenen 2013).

A number of studies have proposed evaluation methods for KBSs. However, it is challenging to decide the best method to use. Some methods are suitable to use with a certain system and not applicable to others. This is understandable, as a KBS has several components: knowledge base, inference engine, user interface, etc. Thus, we can evaluate a KBS based on its components: the knowledge base or the inference engine. More details on the difficulties and variation of methods of KBS evaluation are discussed in the next section.

2.6.1 Philosophy of KBS Evaluation Process

The existing literature on KB evaluation (Batarseh & Gonzalez 2015; Hartung & Håkansson 2007; Knauf, Tsuruta & Gonzalez 2007; Tsai, Vishnuvajjala & Zhang 1999; Vermesan & Coenen 2013) provides various guidelines that mainly focus on measuring the accuracy of the KB after its development. This is typically construed as a standalone stage, called verification and validation. While verification focuses on checking if the knowledge base satisfies the formal specification of the represented knowledge, validation is more concerned with the ability of the knowledge base to reach correct conclusions. These activities are performed sequentially by knowledge engineers to verify that the system has met the user's expectations. Once the knowledge base is verified, the next process will validate it to ensure that the knowledge base will give the correct answer. The validation is done through experiment to ensure that both system's specifications and knowledge base are complete. This model is known as goal checking techniques, and often performed after the system development. The purpose of the evaluation is to collect maximum data concerning to the aim of the study. Often, an evaluation cannot be finished as planned due to complex experimental design or insufficient data.

Unlike the generic knowledge-based systems, the construction of an RDR knowledge base typically begins with knowledge acquisition (Compton & Jansen 1990). The development of a knowledge base starts when the data and information are collected from experts or other sources. This is a sophisticated task in practice such that the reliability of the knowledge base cannot be guaranteed after the acquisition process. Even after the construction of the knowledge base has been successfully completed, the RDR does not provide any formal testing methodology for the resultant knowledge base (Beydoun & Hoffmann 2000b);(Beydoun & Hoffmann 2013). Instead, the evaluation stage typically involves some testing of the dataset over the system. The knowledge base is continued to be built incrementally until it gives an accuracy rate as close to 100% as possible.

The most common method for evaluating RDR-based KBs is by providing a set of test cases with the corresponding known results. Each test case consists of a pair of input queries and expected outputs. In particular, the experts measure performance of the knowledge base against the test cases. The methods for measuring KB performance have varied somewhat across this research area. The first approach concerned with measuring the accuracy of the rule in a KB that represents expert knowledge. This is the most common procedure for determining the performance of a RDR KB, as has been found in many reports (Compton et al. 2006; Galgani, Compton & Hoffmann 2015; Kim et al. 2018; Nguyen et al. 2016). Originally, this process of measuring KB accuracy only involves evaluating the ability of the KB to reach conclusions. However, the drawback of this method is the difficulty to obtain the number of test cases that are actually needed for evaluation.

The second approach is a part of dynamic analysis, in which it requires the analysis to measure the coverage of the KB's rules. This method checks the rules against the test cases and measures the effectiveness of the test cases by analysing the depth of the coverage. The main goal of this approach is to rapidly develop the KB by monitoring the test case execution and calculating the percentage of the KB's coverage against the rest of the test cases. Surprisingly no studies have investigated the coverage analysis-based approach to measure the RDR KB performance. This is a core area of contribution of this thesis. The following sections discuss the dynamic evaluation approach and propose the three steps of dynamic evaluation framework used here in detail.

2.6.2 Difficulties in Evaluating the KBS

There are obvious challenges in evaluating knowledge-based systems, where several factors can lead to difficulties when performing evaluations (Tsai, Vishnuvajjala & Zhang 1999). The first thing that causes the difficulty is the knowledge representation formalism. A KBS has many formal representations of knowledge. The knowledge may be represented as rules, logic, cases, object-oriented models, semantic networks, or generic tasks. A single formalism such as rule-based KB may not cause validation problems. However, the problem may arise when a KB uses multiple formalisms to represent knowledge. For example, (da Silva, Matelli & Bazzo 2014) use several formalisms of rules, object-oriented modelling and semantic networks to represent knowledge.

The second challenge is caused by the characteristics of KBSs that are always evolving in the implementation (Tsai, Vishnuvajjala & Zhang 1999). This concern may increase the size of the knowledge base as well as the number of rules. A classic example of this issue is MYCIN experience, where the KB grew in size and the number of rules increased to 10,000 in 1988 (Barker & O'Connor 1989). The evaluation in this situation could be challenging due inconsistencies, incompleteness, and redundancies. Although the size and number of rules are not always the issues for a KBS, this condition may give rise to evaluation problems with regards to configuration and maintenance (Kang, Gambetta & Compton 1996).

The next issue while evaluating a KBS is caused by the cost of generating, executing, and evaluation of the test cases (Knauf, Tsuruta & Gonzalez 2007). Test cases are required for input testing. They should contain the specification of inputs and expected outputs. A large number of test case are needed to test the KBS. However, it raises a challenge as the input and output space for the test cases can be huge if all criteria in problem specification are considered. On the other hand, selecting the criteria for test case generation could be problematic. Questions that could be raised are related to consistency and completeness. Selecting inappropriate parameters can lead to ending up with test cases that do not reflect the real domain or real problems in the field. If there is no restriction when selecting parameters, the generation of test cases can be difficult and possibly very expensive. Besides the cost of test case generation, the costs of testing also include the cost of evaluation and loss (or fault). Cost of loss increases if the testing fails to identify the fault, and vice versa.

Another concern in evaluating a KBS is selecting the criteria for measuring the performance of KBS (Vermesan & Coenen 2013). There are several criteria for measuring the quality of a KBS, which are, accuracy, error level, reusability, maintainability, and reliability. Of these, accuracy has been mostly used for evaluating the knowledge-based systems. However, determining the standard criteria for evaluating knowledge bases has been a major area of interest within the field of KBS evaluation.

2.6.3 Evaluation Methodologies

Many publications on evaluation of KBSs have been introduced in literature. Each work on evaluation interprets its roles, stages, tasks, and purposes. Deciding on the KBS evaluation method is determined by the problem being addressed and the goal of the evaluation itself (Tsai, Vishnuvajjala & Zhang 1999). Some conditions, like the availability of experts, type of case study, available time and budget, should also be considered before selecting an evaluation method. Several studies have presented guidelines for evaluating knowledge-based systems, which involve understanding of the objects being evaluated (Adelman & Riedel 1997; Rhee & Rao 2008). As illustrated in Figure 2.11, there are three classifications of KBS evaluation: (1) technical aspects; (2) empirical aspects; and (3) subjective aspects. These approaches will be described in the following subsections.

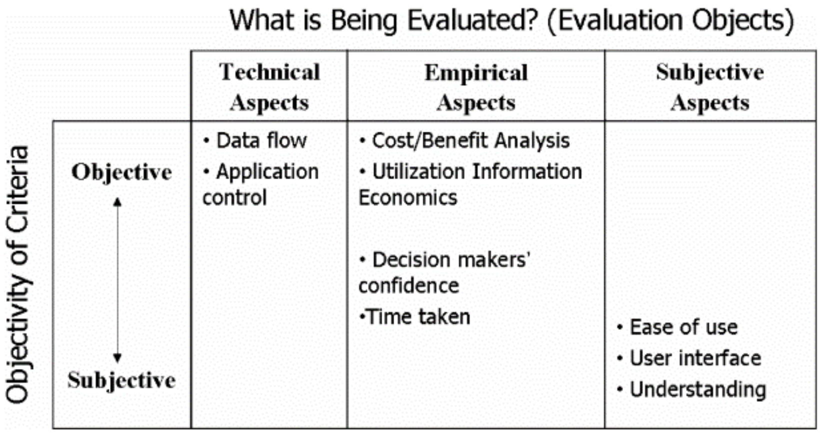


Figure 2.11 Framework for evaluating knowledge-based system (Rhee & Rao 2008, p. 316)

2.6.3.1 Technical Evaluation

The idea behind technical evaluations is to examine the practical aspect of a knowledge based-system. The technical aspects that should be examined are algorithms, data flow, and system logic. The examination generally uses various tools, such as tables, matrices, algorithms, and models. The common evaluation in this category is direct validation of knowledge base, such as analysis of the logic and rule validation (O'Keefe & O'Leary 1993).

Rule validation is carried out by examining the knowledge base directly to inspect the accuracy and validity of the rules. The errors that might be found in rule validation (El-Korany et al. 2000) are (1) wrong Boolean logic in the condition part, (2) incorrect conclusion part, and (3) inconsistent rules with requirements. When the size of a KB is too large for rule validation, experts can perform individual selection of the rules. For instance, evaluation can be made based on the most fired rule in the KB, and vice versa. This approach can be used if the newly added rules have least impact on the KB. If the rules have a certainty factor, then validation can be made based on their weight. Another way to inspect individual rules is by examining the most costly or largest profit rules. The rule validation can affect the performance of the knowledge base, if the allocation of correct representation rules is ignored. An example of this is presented in (Santos Jr & Dinh 2008), where they represent the knowledge base using the Bayesian model. The criteria for validation are uncertainty and incompleteness. They present a probabilistic model extended from the Bayesian network to represent knowledge in an uncertain domain.

The use of metrics was proposed in many academic literatures. In (Ramsey & Basili 1989), the evaluation was conducted to explore the abnormal pattern of the KBS. The metrics used for evaluation were computer runs, time, software changes, and programmer hours. The knowledge base for the comparison purpose was built bottom-up and up-bottom. The experts evaluated and suggested the bottom-up approach with deducting rules for the intended purpose. Different metrics were used in (Menzies 1998). They introduced the use of critical success metric (CSM) for evaluating the KBS without requiring a panel of human experts. The behavioural success criteria from the business case was utilised to measure the success criteria of the system. The CSM was intended for domain-specific KBSs since it analysed the behaviour of the system in the business context. It was also intended for evaluation in the early design of the lifecycle.

In (Nick, Althoff & Tautz 1999), they suggested a practical evaluation using Goal-Question-Metric (GQM), which is based on industrial technology for goal-oriented software measurement. GQM ensures the consistency and completeness of measurements with the structure discussion. In GQM, the evaluation is facilitated by various measurements for each phase of the system development. For instance, acceptance measurement of usage is used during prototyping; formal feedback from user is used for regular usage; and cost-benefit evaluation is used if the usage of the final implementation is wide-spread.

Another approach is using a meta-model (O'Keefe & O'Leary 1993). Meta-model expresses the relationships between the elements of a model that is a model of a model. This validation is useful for a large KB. The availability of constructs and concepts (meta-models) can then be used to determine conceptual validity. The casual model can be used to generate a set of rule bases, and then the experts can check for completeness of the knowledge base. This validation can be used for understanding large rule-bases, but in its implementation it may miss the rule validation if the focus is on casual representation.

2.6.3.2 Empirical Evaluation

Empirical evaluation focuses on the entire performance of the knowledge-based system, with quality of output as the stand point. Thus, this evaluation would require a comparison between outputs of the system with the expectation from experts. Several parties that are involved in the design and implementation stages, should be involved during the assessment. The purpose is to verify the effectiveness, problem solving quality, and hence improve the system. The effectiveness is usually assessed using experimental methods, interviews, case studies, and time series. Among the available methodologies, empirical evaluation is the most popular as it considers both subjective and objective criteria for evaluation.

In (Knauf, Tsuruta & Gonzalez 2007), a framework called validation of knowledge base (VKB) was presented to evaluate and improve the validation result while reducing the time spent of the experts. As shown in Figure 2.12, after the evaluation process, the cases with the best rated solutions are stored. This process can reduce duplicate cases that have already been solved by the VKB. Then, the panel of experts only needs to solve the new case that is not in the part of VKB. This process is collected in a software agent called VESA. Because the VESA already has knowledge from previous experts, this method can be used in the validation process when the experts are not available. There is also the hope that VESA can reduce the availability of human involvement in the validation process.

In (Batarseh & Gonzalez 2013), an incremental knowledge validation framework for KBS based on selective test cases was proposed . As presented in Figure 2.13, the method is named as MAVERICK (Method for semi-Automated Validation Embedded into the Reusable and Incremental Common KADS). The method uses a semi-automated and incremental method that is embedded into CommonKADS. MAVERICK operates based on test case creation and inspection validation. It uses the results from previous validations and iterates to select the next set of test cases. This validation procedure works incrementally. The experiment shows that this method was successful in discovering 94% of the errors (Batarseh & Gonzalez 2013).

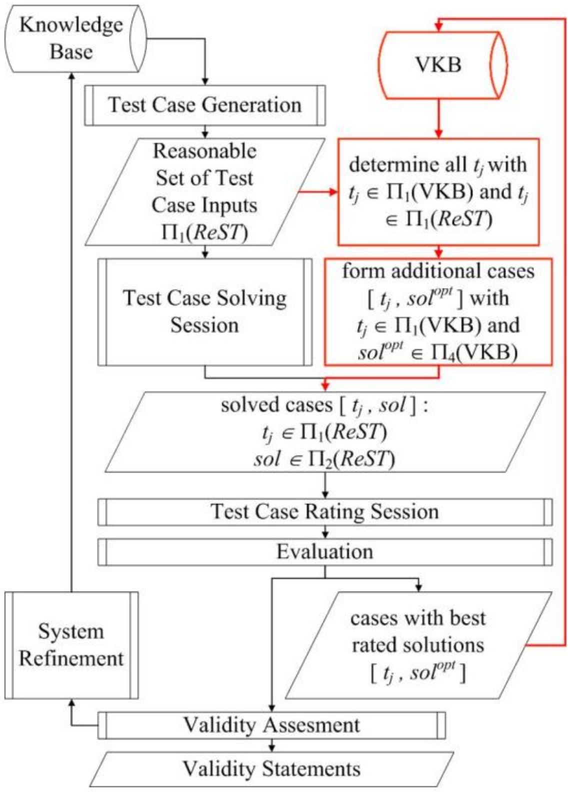


Figure 2.12 Validation knowledge base framework (Knauf, Tsuruta & Gonzalez 2007, p. 122)

In (Onoyama & Tsuruta 2000), the evaluation was conducted by comparing the results of the system with a Turing test. Although a Turing test is suitable for evaluating a knowledge-based system, it is very time consuming for the experts. Hence, they share the validation process among the experts, knowledge engineers, and the system to reduce the time needed by the experts during assessment. This method is known as a Bi-directional many-sided explanation typed multi-step validation method (Figure 2.14). The method is a multi-step validation process to eliminate the centralised knowledge in the system. This is so that the experts can concentrate on the areas that cannot be covered by the system and justify the reason. This is the aim of utilising a bi-directional, multi-step and many-sided validation method.

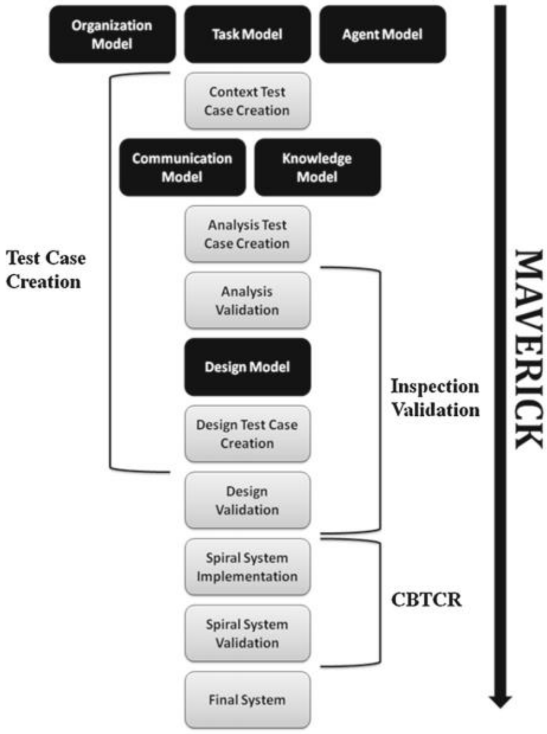


Figure 2.13 MAVERICK Framework (Batarseh & Gonzalez 2013, p. 245)

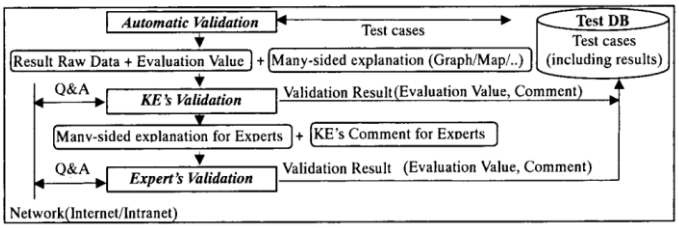


Figure 2.14 Bi-directional many-sided explanation typed multi-step validation method (Onoyama & Tsuruta 2000, p. 464)

Another approach introduced by (Barr 1999) uses a coverage measure to evaluate the testing process. As illustrated in Figure 2.15, the coverage analysis reaches the performance evaluation by comparing expected results with the coverage information and actual results. The crucial component is the coverage information as it includes a comparison of coverage analysis and expected results. The coverage analysis is performed to estimate the degree of incompleteness from the test case and also the potential error from the KBS. Without measuring the completeness and potential error, the system could run well on the test data but contain unmeasured errors in the real test.

Thus, the coverage analysis will give a degree of confidence about the performance of the KBS.

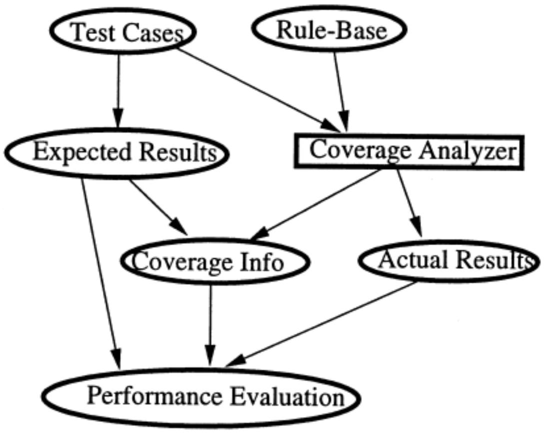


Figure 2.15 KBS Evaluation with Coverage Analysis (Barr 1999, p. 28)

In their study, (Beydoun & Hoffmann 2013) proposed statistical analysis to evaluate the performance of the KB. This process is instantiated for an incremental knowledge acquisition methodology RDR. This approach operates by monitoring the test cases and rule additions in the KB. As a matter of fact, every rule has different impact on the KB status. In case of RDR KB, the initial rules will have a higher impact than the later ones. Rather than evaluating all the rules against the test cases, this method works by monitoring the newly added rules and estimates the performance. The parameters of the performance include coverage and predictivity. In the end, this validation method aims to estimate the decision when the running test cases can be terminated. The research presented in this thesis makes a theoretical contribution by implementing this approach for evaluating an adaptive e-learning system KB. Thus, the method will be discussed in greater detail in Section 5.

2.6.3.3 Subjective Evaluation

Subjective evaluation measures the effectiveness and usefulness of knowledge-based systems for the user, organisation, and environment. The assessment includes the examination of the system to address the problem, how systematic its problem-solving techniques are, and whether it could resolve the requirements of the end-users. The criteria for the evaluation is subjective with the purpose of understanding the usability and interfaces of the implemented system. Thus, this evaluation is usually domain-dependent with the specific purpose of evaluating the acceptance of the end product to the end user.

In (Hussain, Wardman & Shamey 2005), a KBS to diagnose the faulty dyed samples was developed. In order to evaluate this system, several evaluation copies of the KBS were distributed to the targeted users. The purpose of this measurement was to measure the end user opinion for the value and utility of the system. The selected users were to critically evaluate the system in terms of its interactivity, task simplification, user-friendliness, speed, ease of understanding, expert helpfulness, and ease of use. The evaluation criteria was presented in a scale number from 1-10. The average scores from the participants represented the likelihood that this system would be accepted by the community.

In a recent study, (Abdullah, Ligęza & Zafar 2017) proposed a performance analysis of a knowledge-based expert system based on the measurement of time spent for processing a medical claim. The system was intended for handling the medical billing process and used a rule-based method for constructing the knowledge base. The system helped to auto-correct the faulty claims. During the evaluation, a statistical analysis was used to examine the improvement of the system based on time spent to handle a case.

2.7 Chapter Summary

This chapter reviewed the existing related works that are presented in this thesis. First, the body of research on adaptive e-learning system was presented. This covered the requirement and challenges, the conceptual models, and the general approaches for modelling the adaptive learning system. Historical background on knowledge-based systems was introduced. Existing works and applications were reviewed and then the structure of the KBS was discussed. This description was followed by the analysis of Ripple-Down Rules that focused on the philosophy underpinning the RDR, existing approaches and variations, strengths and limitations, and also the recent applications of Ripple-Down Rules. The literature review showed that, while some works have suggested mechanisms to apply RDR on various models, none have implemented the method to solve problem of adaptive learning. This thesis is set out to investigate this gap.

Despite this promising opportunity to advance our knowledge of examining RDR for adaptive learning, there are still unanswered questions about evaluating the knowledge-based system. The literature review revealed that evaluation was often a difficult part of the KBS. In particular, this thesis examines dynamic evaluation for tackling KBS validation. In the next chapter, the thesis will present the design of research that leads to propose a mechanism to evaluate the KBS. The domain will be represented as an adaptive e-learning system and two case studies will be used to evaluate the initial model and its ability to solve adaptation problems.

3 RESEARCH DESIGN

The previous chapter has presented a literature review used in this thesis. In particular, it discussed the methods and best practices for developing a knowledge-based system. Although the interest in this field is still widely-investigated, only a limited number of studies focus on evaluating the KBS while developing it. The focus of this research, therefore, is to present an evaluation of the KBS to the next level by monitoring its development during acquisition process. This chapter aims to elaborate on the process that underpins the research by discussing the research methodology. The remainder of the chapter is organised as follows. Section 3.1 reviews the rationale of the design science research methodology. It will then go on to describe the four phases adhered to in this research: problem identification is presented in Section 3.2; the design of a learner model for adaptive e-learning systems is introduced in Section 3.3; initiating the conceptual framework is discussed in Section 3.4; and evaluation of the concept will be presented in Section 3.5. The final Section 3.6, will summarise this chapter.

3.1 Overview of Design Science Research Methodology (DSRM)

As mentioned in the first chapter, this research proposes a new method that enables rapid evaluation of the developing KBS. This research uses a Design Science approach to understand the way applied systems solve the problem. By definition, design science is "*the design and investigation of artefacts in context*" (Wieringa 2014). An artefact in Design Science Research (DSR) can be represented in various forms (see Table 3.1), such as construct, model, frameworks, architecture, design principle, methods, instantiation, and design theory (Hevner et al. 2004; Vaishnavi & Kuechler 2004; Wieringa 2014). In DSR, these developed artefacts are designed to interact with the domain problem to solve the problems or improve the context. The process of design science, as highlighted by Vaishnavi & Kuechler (2004), encompasses the construction of new knowledge through an original design of artefacts (things or products). Furthermore, the designed artefact (s) needs to be analysed for both, the usage and the performance, to understand and enhance the role of the artefact in the context of the information system.

Table 3.1 Artefacts of Design Science Research (Vaishnavi & Kuechler 2004)

Artefact	Description
Constructs	The symbols or vocabulary to communicate a research domain
Models	The abstractions that express the relationships between constructs
Methods	Sets of steps (e.g. algorithms, process) that provide a guide to performing the tasks
Frameworks	The conceptual guides which serve as a guidance
Architectures	High level structures that underlying the designed systems
Instantiations	The proof of concepts of an artefact in its environment
Design Principles	The main concepts and principles that guide the design of the artefact
Design Theory	Sets of statements that facilitate understanding of the guidance to achieve a certain objective

Design science is elected as the methodological research approach in this research because of two factors. Firstly, design science is a research method which focuses on practical problem solving with continuous improvement. This step needs solution-oriented knowledge, where the expected output is the reasoning of science that can be used to solve a complex and relevant problem. Since design science is field-problem driven and is solution-oriented, it is better suited for the implementation of this approach in the domain of AES. Secondly, many types of research are based on descriptive knowledge (explanatory science) in the domain of AES. The model of this research is to use theory to develop knowledge by understanding a phenomenon or specifically, explaining and predicting knowledge.

Though design science roots are in engineering and computer science, they can also be found in many other disciplines and fields. In the information system domain, the research showed the thriving diversity of the approaches, methods, and techniques from technical to organisational issues (Gregor & Hart 2010; Myers & Venable 2014; Peffers et al. 2007). For instance, Wieringa (2014) argues that design science has five phases of design cycle as follows:

1. Problem investigation. Explain what a phenomenon is and why it needs to be investigated and improved.
2. Treatment design. It is related to the design of one or several artefacts in solving the

problem.

3. Treatment validation. Explain if a design can solve a problem.
4. Treatment implementation. Treat the problem with one of the design artefacts.
5. Evaluation. Can the treatment solve the problem? What should be improved, and should the cycle start from the beginning?

In another study, Hevner et al. (2004) suggested well-known seven guidelines to conduct DSR in the Information Systems field. Table 3.2 summarises the seven guidelines of conduct, design evaluation and communication of research as proposed by Hevner et al. (2004). This is also followed by the descriptions of the applications in this research.

Table 3.2 DSR Guidelines (Hevner et al. 2004, p. 83)

No	Guideline	Description	Application in this research
1)	Design as an Artifact	DSR needs to produce an applicable artefact	The development of artefacts that includes constructing, model, method and architecture
2)	Problem Relevance	Artefact in DSR is developed to provide important solutions that are relevant to human problems	An incremental knowledge acquisition method will be adopted to support recommendation of learning objects in AES
3)	Design Evaluation	The artefact needs to be demonstrated through rigorous evaluation procedures in terms of quality and efficacy	The design artefact will be evaluated through the two case studies
4)	Research Contributions	DSR must provide a clear contribution in terms of the design or methodologies	The design artefact will contribute a new understanding of evaluating a developed KBS
5)	Research Rigor	The design artefact must be rigorously constructed, implemented, and evaluated	The research rigour is achieved by testing and evaluating the design artefact to determine the best solution with the reasons to improve the design.
6)	Design as a Search Process	The expected artefact is achieved from an effective searching process while satisfying the problem	The design artefact has its own methodology that will be applied to improve the knowledge of the framework.
7)	Communication of Research	The output of DSR should be effectively presented to technology-oriented and management-oriented audiences	The communication of research were presented through conference and journal publications (Muhammad et al. 2018; Muhammad et al. 2016)

The implementation of DSR in this research is organised in four phases. These phases were created in accordance with the DSR guidelines in Table 3.2. As illustrated in Figure 3.1, the first phase focuses on identifying the knowledge gap from relevant literature. This is the implementation of design as an artifact and problem relevance. Then, the realisation of design evaluation is provided in Phase 2, designing the generic learner model for adaptive e-learning. Following this in Phase 3, the design and development process that addresses the question of the research is examined. This phase is the realisation of design evaluation and research contribution. Finally, Phase 4 demonstrates and evaluates the proposed prototype. This is the implementation of research rigor and design as a search process. According to the principles of design science research, following evaluations, prior phases are revisited to refine the model and the knowledge base development process.

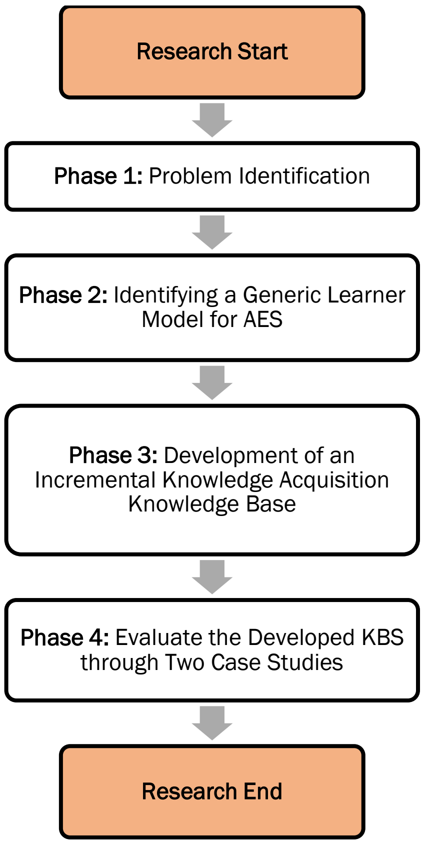


Figure 3.1 Research Phases

3.2 Phase 1: Problem Identification

The first phase in this thesis is identifying the problem and exploring opportunities in the current environment. In this stage, existing knowledge is investigated to raise the main problem in this project. In DSR, this phase is also known as problem investigation, where the purpose is to understand the underpinning problem and explore a better designed solution. According to Wieringa (2014), there are four categories to investigate the problem: problem-driven, goal-driven, solution-driven, and impact-driven investigation. These processes of problem identification are similar to those suggested by Peffers et al. (2007). As shown in Figure 3.2, there are four possible entry points in DSRM: problem-centred initiation, objective-centred solution, design and development-centred initiation, and client/context initiation. By these classifications, the problem identification process may start at any step and move outward.

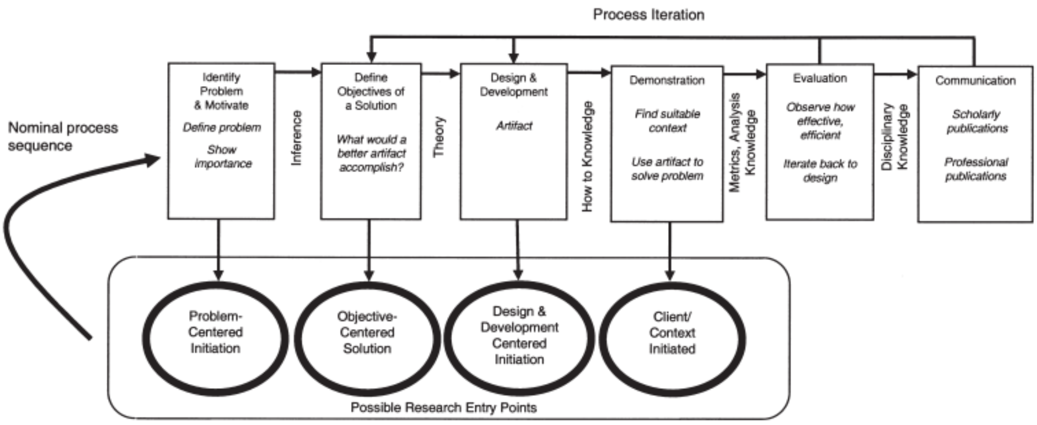


Figure 3.2 DSRM Process Model (Peffers et al. 2007, p. 54)

The problem investigated in this research fits into *problem-centred initiation,* which is known by reviewing previously published studies. The review initiates by studying relevant research related to knowledge acquisition and knowledge-based systems, specifically related to evaluation, validation, verification, issues and opportunities. From the review of the literature, it is understood that most studies in this field have only focused on the designs and implementations. Little is known about the evaluation process of KBS, and it is not clear when the evaluation process usually completes. Only a limited number of studies have been conducted to address this problem. In this thesis, the evaluation field was surveyed, and we focused on dynamic evaluation by monitoring the knowledge acquisition process (Beydoun & Hoffmann 2013) for determining the conditions to terminate the KBS development process. Despite this promising philosophy, more investigation needs to be undertaken as the approach has never been applied in practice. After reviewing numerous models in the knowledge-based system, the rule-based model using knowledge acquisition Ripple-Down Rules was selected after considering its flexibility and ease of decision making. Then, a knowledge-based system was proposed as the artefact to provide a rich environment of AES that enables improvement of the research methodology on validation issues.

After developing a proof-of-concept-level prototype, this research confirmed the capability of the proposed KBS to support and automate the e-learning system. The artefact was implemented under two case studies with data classified into three models, which are:

1) Learner model to express an abstract overview of a learner in AES.
2) Domain model that contains learning resources with structure and descriptive information.
3) The instructional model which simulates the adaptation by applying selection rules.

Then, the evaluation was conducted for obtaining the optimum condition of the knowledge acquisition process. The statistical analysis took into account the validation process.

Chapter 1 and 2 of this thesis are the output of the activity in phase 1. Chapter 1 provides the rationale behind this research, ranging from the description of the problem to the solution of this research. While Chapter 2 is the description of the investigation problems that exist in the literature.

3.3 Phase 2: Identifying a Generic Learner Model

Following the DSR paradigm, the second phase produces an artefact to represent the AES domain, as identified in Phase 1. The artefact output of this phase is the learner model. The purpose of identifying the learner model is to represent the domain knowledge of a learner in an adaptive learning environment. The representation of the domain knowledge is designed to be generic so that a course from any domain of AES can be configured into the system. The implementation of a generic learner model is adapted from (Beydoun, Low, Mouratidis, et al. 2009) by eliminating the last step. The overall development process of the generic learner model in the second phase is illustrated in Figure 3.3. The output of Phase 2 will be described in detail in Chapter 4 of this thesis.

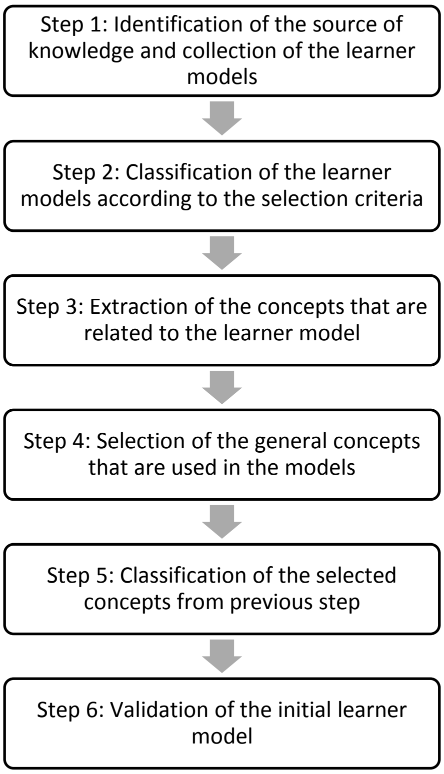


Figure 3.3 Overall Guideline of Learner Modelling Process

The above Figure 3.3 is a guideline for constructing a learner model in AES. It has six steps where each step contains detailed instructions on the tasks and processes that should be performed. A brief description of each step is as follows:

1) **Step 1: Identification of the source of knowledge and collection of the learner models**. The main activity in this step is to collect models from several knowledge sources. To obtain the best results, the source of knowledge is elected from a reputable online database, including EBSCO, Scopus, and Web of Science. Then, a set of criteria is used to search and select the desired articles.

2) **Step 2: Classification of the learner models according to the selection criteria.** The results from the previous step are classified according to the selection criteria. For our research, three selection criteria were used: Personal User Characteristics;

Previously Acquired Knowledge and Skills; and System-Related User Characteristics. These classifications were selected as they contain the general taxonomy for a learner model. Thus, the candidate models were classified based on their completeness and coverage.

3) **Step 3: Extraction of the concepts that are related to the learner model.** The extraction process involves the identification of concepts among models that have high feasibility to be included in the initial learner model. The extraction process is carried out by inspecting all the articles, then identifying the important concepts to be used as model data. The extracted concept should represent things or objects from the learners.

4) **Step 4: Selection of the general concepts that are used in the models.** In this step, the list of concepts derived from the previous step was analysed and refined. The first step in selecting a candidate concept is by filtering the number of occurrences and generality of the name. The same concept in each model is counted, then sorted from highest to lowest occurrence. Concepts that appear in at least three models are selected as candidates for inclusion in the model list, while the concept that appears only in one or two models is discharged from the process.

5) **Step 5: Classification of the selected concepts from previous step.** In this step, the selected concepts are grouped into categories according to their function in the learner model. As stated in Step 2, the classification is selected from the review of studies based on the general taxonomy of the learner model on AES.

6) **Step 6: Validation of the initial learner model.** The last step is validation to measure the quality of the learner model by investigating the generality, expressiveness and completeness of the proposed learner model. For this purpose, the validation process follows frequency-based selection as the validation technique.

3.4 Phase 3: Development of an Incremental Knowledge Acquisition Knowledge Base

The activity in Phase 3 is the design and development of the main artefact, which is an instantiation to solve the problem as identified in the first phase. The instantiation is realised as a knowledge-based system that supports adaptive learning. The prototype is constructed from scratch in the form of a web-based program. This thesis utilises Apache web server, PHP server script language, and MySQL database server for prototyping. The knowledge base in the system is built up incrementally from the test cases supplied into the system. The system refines the test cases and builds rule tree based on the cases.

There are two requirements that need to be filled for the development process. First, the system should facilitate knowledge extraction and structuring of the knowledge from sources. The system does not allow direct rule creation nor correction to the existing rules as this could negatively affect the existing knowledge in the KB. All rules will be created automatically from errors that are found during the knowledge acquisition process. However, the role of the expert is still needed to perform manual refinement. The second requirement is that the system should facilitate the monitoring of the knowledge acquisition process. Thus, a monitoring module is developed to calculate statistical analyses on the fly during the knowledge acquisition process. The monitoring module will present the performance of the KB for further analysis. The output of the artefacts which include model, framework and architecture, are presented in Chapter 5.

3.5 Phase 4: Testing and Evaluation of the Concept

The main activity in Phase 4 is to evaluate the artefact developed in the previous phase. The evaluation in design science methodology has several goals (Johannesson & Perjons 2014). The main goal of the evaluation is to determine the extent to which the artefact is effective for solving the proposed problem. Besides, an evaluation can also have other goals, such as to evaluate the requirements, investigate formalised knowledge, investigate side effects, and identify opportunities for improvement. The evaluation goals can be combined. As in this research, the goal of evaluation is to investigate formalised knowledge and obtain information to improve the design.

As mentioned before, the instantiation in this research is a KBS to support adaptive learning. The first activity for the evaluation is to evaluate the first requirement, that is, an investigation on how well the artefact solves the problem of adaptivity in AES environment. It is demonstrated by implementing a real course of study within the e-learning system. The course of study chosen for the first case study is "Web Development", while the topic in the second case study is "Networking Essentials". This will enable the implementation process for different presentation styles and illustrate on how to achieve adaptivity. It will also illustrate that the approach in guiding the learning path of a student can be used in different topics of learning.

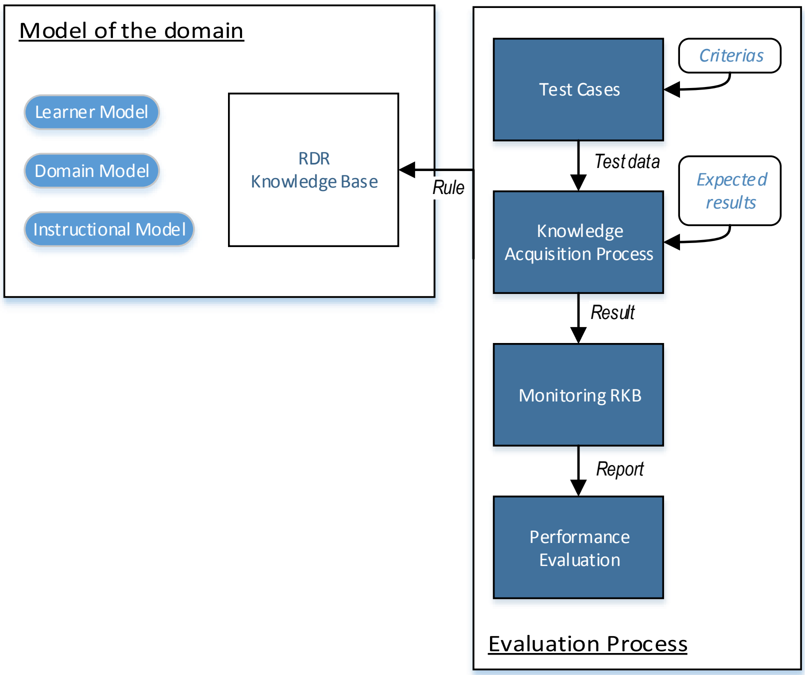


Figure 3.4 Guideline for Evaluation Process

The second requirement is to evaluate the KB by monitoring the knowledge acquisition process. Figure 3.4 illustrates the guideline for evaluating the process. The first activity is to generate and optimise a set of test cases that consist of a combination of input and expected outputs. The test cases will simulate the actual operation of the knowledge base. The test data should represent the coverage of all possible combinations with at least a pair of attributes represented in the test cases. Therefore, pairwise combinatorial testing is adopted in this process.

Then, the evaluation analysis process runs on the fly and parallel during the knowledge acquisition process. This research adopts statistical analysis to evaluate every change during the knowledge base update. The modification of RDR-knowledge base during the knowledge acquisition process is monitored and integrated with the statistical analytics. Statistical analysis is a way to estimate the coverage of the knowledge base and predict the certainty level of new rules. Overall, this activity aims to evaluate the coverage of the rules and measure the predictivity from the knowledge base decision. The trend of the analytical result will determine if the knowledge acquisition process should be stopped or continued. The final results will conclude the performance of the developed knowledge base.

3.6 Chapter Summary

This chapter explores the research methodology used in this thesis. A design science approach is followed to create an adaptive learning system based on an incremental knowledge acquisition approach. The design and development of the AES will be interleaved with the evaluation by monitoring the KB readiness. The research is structured in four iterative phases. The first phase is problem identification. The problem investigated in this research fits into the category of design and development-centred approach, where this research enhances and further develops the existing adaptive learning systems. Phases 2 and 3 refer to the artefact development based on the problem identified in the first phase. Phase 2 is the development of the generic adaptive learning model, follow by the development of adaptive learning system based on incremental knowledge acquisition in Phase 3.

Phase 4 covers the evaluation phases of the artefacts. All these phases are elaborated in the corresponding chapters in this thesis. The problem identification from existing research was compiled in Chapter 1 and 2. The research methodology for answering the research gap is reviewed in Chapter 3. The development of an adaptive learning metamodel will be explored in Chapter 4. The development of the proposed model will be demonstrated and illustrated in Chapter 5. Once the model is developed, the next process to be followed is the evaluation. In this research, the purpose of the evaluation process is to estimate the effectiveness of the developed knowledge base. The final conclusion and potential future work will be presented in the last chapter, Chapter 8.

4 IDENTIFYING THE LEARNER MODEL OF ADAPTIVE E-LEARNING SYSTEM

As described in Chapter 3, the topic in this thesis covers two significant areas: identifying and fulfilling modelling requirements and development of adaptive e-learning systems. The first part is described in this Chapter, while the second part is discussed in Chapter 5. The modelling requirements in Adaptive E-Learning Systems focus on two issues. First is identifying the learner model for adaptive e-learning development. The aim of modelling the learner is to answer the initial question of adaptivity: what characteristics of the users should the learner model describe to enable the e-learning system to assist them in attaining high achievements? Before defining and presenting the characteristics of the learner, it is necessary to have a good conceptual understanding of the fact that adaptive learning has a different aspect too. The second issue is how a learner and the elements in their environment can be jointly analysed and exploited to provide useful information about the learning process that is constructed from each learning episode.

This chapter presents a modelling technique to identify a set of learner models based on a rigorous literature review. It begins by presenting the procedures to extract the learner model. It creates the initial version of the first component of the design science artefact that this research aims to produce, the learner model. This model describes the concepts commonly used in modelling students and their environment in adaptive learning. The procedure presented in this section portrays the adaptive e-learning system from the angle of a student. The purpose is to identify the essential properties that characterise the student in an e-learning environment. Section 4.2 presents the definition of each feature in the learner model and their potential values that represent the characteristics of the learner. Finally, concluding statements to further justify the learner model are presented in Section 4.3 to conclude this Chapter.

4.1 Procedure to Construct the Learner Model

This chapter discusses the specific method of how the learner model was constructed. The main focus of this section is to find future aspect of the learner's model from the relevant literature review. First, the procedure to extract the concept of the learner model is described, followed by a presentation of the proposed concept. The modelling procedure is adapted from the metamodeling process of introduced by Beydoun, Low, Henderson-Sellers, et al. (2009). It is a rigorous and well-defined method that lends itself to reuse. As shown in Figure 4.1, the adaptation in the methodology includes additional model collections of AES models, exclusion of the relationships between concept, and inclusion of the validation methods in the last step. Specifically, the goal of each step is the following:

- Step 1: Identification of the source of knowledge and collection of the learner models.
- Step 2: Classification of the learner models according to the selection criteria.
- Step 3: Extraction concepts that are related to the learner model.
- Step 4: Selection of general concepts that are used in the models.
- Step 5: Classification of the selected concepts from the previous step.
- Step 6: Validation of the initial learner model.

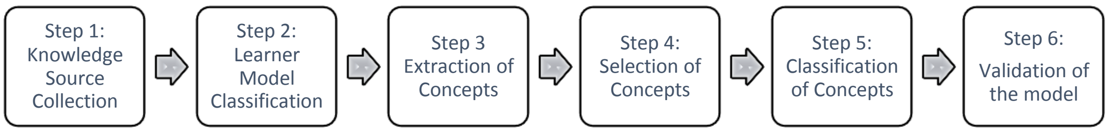


Figure 4.1 Learner Model Creation Process

4.1.1 Step 1: Identification of the source of knowledge and collection of the learner models

The first step in the refinement process is to collect models from the knowledge source. For the sake of ensuring identification of major contributions in leading journals, a model search was undertaken using reputable online databases including EBSCO, Scopus, and Web of Science. A set of criteria was used to search and select the articles. If the resulted papers did not meet the selection criteria, then they were excluded from the collection. This research used three criteria which are described as follows:

- The literature was based on a set of keywords search in the topic fields (title, abstract, and keywords) of publications. The search phrase was composed of three fragment joints with the AND operator. The first fragment covers the adaptation and personalisation, so the term (adapt* OR personal*) was used as the first keyword of the search phrase. The second fragment of the search phrase is to cover the field of e-learning and educational system. The second fragment consisted of the following terms: (e-learning OR hypermedia OR "web-based learning" OR "intelligent tutoring"). Then, the third segment restricted the search only on a paper reporting the learner model in their framework, so the term ("learner model" OR "student model" OR "user model") was used. The last fragment was used to focus only on the research that had been evaluated or implemented in an educational setting. So, the search terms (evaluate* OR empiric* OR experiment*) were applied as the last fragment.
- Along with the keyword fragment, the search literature was limited to the articles published in the English language from 2005 to 2017. The consideration for selecting this period is to get the latest state of the art research trend in this domain.
- The final criteria of the literature search were to survey only journal articles and book sections. Therefore, other publication forms, including conference proceedings, books, technical reports, or unpublished working papers were not included in the selection.

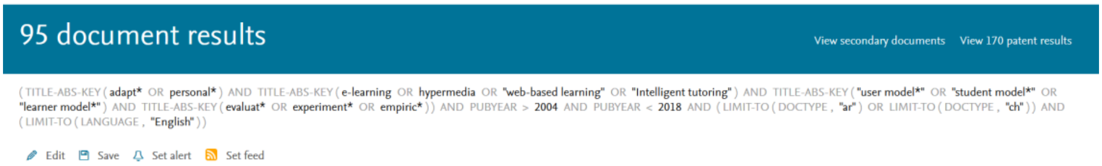


Figure 4.2 The search result of the model in Scopus

As can be seen from figure 4.2, a search with the proposed criteria using Scopus has produced 95 document results. By performing a similar search procedure to other databases, the total number of articles found was 145 papers. Most of the articles were published in the User Modelling & User-Adapted Interaction journal with twenty-six papers, followed by Computer & Education and IEEE Transactions on Learning Technologies with eight papers each. Following this, the abstracts and full texts were thoroughly inspected. After reading all available accepted articles, a total of 82 models were selected for further analysis.

4.1.2 Step 2: Classification of the learner models according to the selection criteria

The selected learner models from the previous step were collated in a single table to ensure that every title only appears once. The selected models were refined based on their completeness in this step. The criteria of completeness refers to the coverage presented by the model concerning the types of learner model as suggested by (Granic 2008; Granić & Nakić 2010): 1) *Personal User Characteristics;* 2) *Previously Acquired Knowledge and Skills;* 3) *System-Related User Characteristics.* These classifications were selected as they contain the general taxonomy for learner model. Therefore, the designation could be applied to many models and was not specific to certain models.

As presented in Table 4.1, 33 models were found according to the criteria. These models were distributed in two different sets, Set 1 and Validation Set 1. The process of refining the model started from identifying the high level and general concept. The models in Set 1 were used as the main source of extraction. These sets contained the 23 learner models that were covering at least one type of the classification presented above. The models were selected based on the coverage and the number of works cited in the articles. Some papers cover all type of learner models, while others only focus on one type or none. The articles that did not cover at least one type of learner model were discarded from further processes. As the task of the first step is finding the common vocabulary, many references meet the criteria. Then the process went details afterwards by refining the candidate concepts for its occurrence and consistency. This process is exhaustive and time-consuming. However, the process was reliable and rigour from the identification of the concepts to validation.

Table 4.1 Model classification

No	References	Total Cited	Year Published	Personal User Characteristics	Previously Acquired Knowledge and Skills	System-Related User Characteristics
Set 1 for the model development						
1	(Chen, Chang & Wang 2008)	304	2008	x	x	x
2	(Sangineto et al. 2008)	139	2008	x	x	x
3	(Martins et al. 2008)	128	2008	x	x	x
4	(Germanakos et al. 2008)	63	2008	x	x	x
5	(Jun-Ming et al. 2006)	63	2006	x	x	x
6	(Jeremić, Jovanović & Gašević 2012)	59	2012	x	x	x
7	(Kim, Lee & Ryu 2013)	50	2013	x	x	x
8	(Reategui, Boff & Campbell 2008)	50	2008	x	x	x
9	(Kelly 2008)	45	2008	x	x	x
10	(Chrysafiadi & Virvou 2012)	38	2012	x	x	x
11	(Papanikolaou 2015)	15	2015	x	x	x
12	(Colace, De Santo & Greco 2014)	12	2014	x	x	x
13	(Peña Ayala 2010)	12	2010	x	x	x
14	(Eberle, Schwarzinger & Stary 2011)	11	2011	x	x	x
15	(Müller & Kohlhase 2009)	9	2009	x	x	x
16	(Ouf et al. 2017)	8	2017	x	x	x
17	(Medina-Medina, Molina-Ortiz & García-Cabrera 2011)	5	2011	x	x	x
18	(Dogan & Dikbıyık 2016)	2	2016	x	x	x
19	(El Bouhdidi, Ghailani & Fennan 2013)	2	2013	x	x	x
20	(Lamia & Tayeb 2013)	1	2013	x	x	x
21	(Hafidi & Bensebaa 2013)	0	2013	x	x	x
22	(Marković et al. 2013)	19	2013	x	x	
23	(Legaspi et al. 2008)	9	2008	x	x	

Validation Set 1 for validation						
1	(Limongelli et al. 2009)	95	2009	x	x	x
2	(Verginis et al. 2011)	5	2011	x	x	x
3	(Jeremic, Jovanovic & Gasevic 2009)	80	2009	x	x	x
4	(Karampiperis et al. 2006)	26	2006	x	x	x
5	(Brinton et al. 2015)	23	2015	x	x	x
6	(Mahnane, Laskri & Trigano 2013)	17	2013	x	x	x
7	(Chrysafiadi & Virvou 2013a)	16	2013	x	x	x
8	(Hoic-Bozic, Mornar & Boticki 2008)	5	2008	x	x	
9	(Scott, Soria & Campo 2017)	3	2017	x	x	x
10	(Colace et al. 2016)	0	2016		x	x

After selecting the articles for Set 1, the remaining ten articles were grouped in the Validation Set 1 and used for validating the initial learner model. The validation process has two objectives: to find the missing concept and to evaluate the importance of the concept. The list of concepts within the validation set 1 are compared with the initial model. If a concept is not found in the original model, then the concept is included as the candidate concept. Furthermore, the calculation of the frequency of occurrence of each concept within the set model is used to evaluate the importance of the concept. If the concept is rarely used, then it will be removed or replaced with other concepts that meet the criteria.

4.1.3 Step 3: Extraction of the concepts that are related to the learner model

The extraction process involves the identification of concepts among models that have high feasibility to be included in the initial learner model. The extraction process is done by inspecting all the articles, and then identifying the important concepts to be used as model data. The extracted concept should represent things or objects from the learners. A common way to identify the things or objects is by finding the noun within the sentence (Ali et al. 2017). For instance, "*Gender, Learning Style, Personal Data*" is a noun and might be included in the initial concepts. The concepts can be extracted from various sources, including through the descriptions, representation of data tables or graphics of each model. For example, Jeremić, Jovanović & Gašević (2012) used the tree diagram to define their learner model, which consisted of personal data, performance data, teaching history and test history. Concept extraction can also obtained from the description of textual models in the paper. As an illustration, Eberle, Schwarzinger & Stary (2011) describes their learner model within the text description. Thus, we can extract several main concepts from the description, such as *Knowledge, Interest, Goals, Preferences, Learning Style, Content, Domain Concepts, Topic Level,* and *Tasks*. The list of previously identified concept could also help facilitate the extraction of the concepts.

The concept extraction started from Set 1 in Table 4.1. Then, the process involved identifying the candidate concepts that could be included in the learner model. The clear and widely accepted definition is strongly considered. The identification and extraction of the concepts were collected manually for each paper. The result of the extraction process is shown in Table 4.2 below.

Table 4.2 Shortlisting candidate concepts

No.	Reference	Candidate Concepts	Total
1	(Chen, Chang & Wang 2008)	Learning Behaviour, Self-Evaluation, Testing Result, Schedule, Learning Condition, Learning Activity, Learning Goal, Motivational Information, Instructional Plan, Course Schedule, Learning Performance, Learning Task, Learning Strategy, Duration, Concept, Difficulty, Learning Device, Place, Presentation, Interaction, Test Result, Announcements	22
2	(Sangineto et al. 2008)	Cognitive State, Domain Concept, Knowledge Degree, Number of Test, Learning Style, Learning Preference, Learning Path, Educational Context, Age, Languages, Learning Styles, Presentation, Classification, Educational, Learning Resource Type, Teaching Style	16
3	(Martins et al. 2008)	Personal Information, Demographic Data, Age, Academics Background, Qualifications, Knowledge, Deficiencies, Domain of Application, Learning Style, Cognitive Capacity, Traces of Personality, Inheritance of characteristics, Objectives, Planning, Navigation, Knowledge Acquired, Results of Evaluations, Context Model, Aptitude, Interest, Deadline Extended, Psychological Profile	22
4	(Germanakos et al. 2008)	Habits, Location, Time, Security, Authentication, User Segmentation, Content Identification, User Perceptual Characteristics, Knowledge, Goals, Background, Experience, Preference, Learning Strategy, Activities, Age, Gender, Device, Bandwidth, Display, Connectivity, Interface, Data Entry, Working Memory, Learning Style, Visual and Cognitive, Cognitive Task, Cognitive Processing, Emotionality	29

5	(Jun-Ming et al. 2006)	Learning Activity, Learning Behaviours, Learning Portfolio, Learning Path, Preferred Learning Course, Grade of Course, Learning Sequence, Learning Time, Gender, Age, Education Status, Learning Experience, Learning Motivation, Learning Pattern, Identification, Course Module, Student Characteristics, Cognitive Style, Learning Style, Media Preference, Social Status	21
6	(Jeremić, Jovanović & Gašević 2012)	Personal Data, Username, Country, Organization, Language, Performance Data, Time of Last Session, Detail Level, Session Number, Programming Language, Teaching History, Concept History, Test History, Unit History, Time, Detail Level, Learning Style, Initial Skill Level, Experience Level, Actual Skill Level, Duration, Number of Passes, Degree of Mastery, Knowledge Level, Instructional Plan, Concept Title, Difficulty Factor	27
7	(Kim, Lee & Ryu 2013)	Experience, Background, Goals, Knowledge Level, Preference, Interest, Interaction Style, Attitude, Learning Experience, Personality, Cognitive Style, Learning Performance, Time Taken, Correct Answer, Trait	15
8	(Reategui, Boff & Campbell 2008)	Demographic Feature, Affective State, Age, Gender, Course, Hobby, Topics of Study, Emotion, Mood, Interpersonal Stance, Attitude, Personality Traits, Knowledge, Login Time, Language, Emotional	16
9	(Kelly 2008)	Knowledge, Background, Learning Behaviour, Navigation, Presentation Strategy, Choice Level, Activity Level, Activity Group, Pedagogical Strategy, Learning Performance, Learning Gain, Learning Activity, Preference, Navigation History, Time, Instructional Design, Intelligence	17
10	(Chrysafiadi & Virvou 2012)	Level of Knowledge, Needs, Characteristics, Misconceptions, Individual Traits, Cognitive State, Learning Performance, Domain Concept, Percentage of Error	9
11	(Papanikolaou 2015)	Pedagogical Duration, Pedagogical Type, Level of Performance, Tools, Learning Style, Knowledge Level, Social Interaction, Instructional Strategy, Cognitive Purpose Indicator, Working Style, Learner Control, Visit on Resource, Sequence of Resource, Time Spent, Action, Type of Assessment, Progress, Social Purpose Indicator, Scope, Learner Effort	20
12	(Colace, De Santo & Greco 2014)	Learning Characteristic, Cognitive Skill, Intellectual Ability, Intention, Format, Bandwidth, Interactivity, Difficulty Level, Time, Learning Style, Knowledge Level, Preference, Competence, Evaluation Test	14
13	(Peña Ayala 2010)	Psychological, Knowledge Domain, Gender, Age, Level of Study, Academic Status, Occupation, Hobby, Home Address, Level of Knowledge, Prior Knowledge, Acquired Knowledge, Background Domain, Personality Attributes, Learning Preferences, Cognitive, Topic, Content Experience	18
14	(Eberle, Schwarzinger & Stary 2011)	Background Knowledge, Interest, Goal, Preference, Learning Style, Content, Domain Concepts, Topic Level, Tasks, Navigation, Presentation, Interface, Cognitive Style, Global Objective	14
15	(Müller & Kohlhase 2009)	Age, Language, Preference, Background, Interest, Behaviour, Social Network, Default Notation Context, Context Parameters, Date, Time, Device, Location, Area, Audience, Task, Event, Expertise, Layout	19
16	(Ouf et al. 2017)	Personal Information, Contact, Demographic, Biographic, Qualification, Certification, Performance, Preference, Format Presentation, Language, Media Type, Interest, Learning Goal, Background Knowledge, Learning Style, Learning Activity, Prior Knowledge, Portfolio, Security, Relation, Competency, Knowledge, Skill, Ability	24

17	(Medina-Medina, Molina-Ortiz & García-Cabrera 2011)	Personal Data, Access Password, Name, Gender, Age, Occupation, Knowledge, Number of visits, Degree of Knowledge, Experience, Navigation, Preference, Knowledge subdomain, Interesting Concept, Knowledge Goal, Location, Task, Language, Level of Difficulty, Media	20
18	(Dogan & Dikbıyık 2016)	Personal Information, Username, Date of Birth, Gender, Start Date, Last Visit, Status, Time, Learning History, Difficulty Level, Course Content, Concept Error Rate, Success Status, Concept Learning Level, Related Concept, Previous Concept, Needs, Misconceptions, Cognitive Ability, Behaviour	20
19	(El Bouhdidi, Ghailani & Fennan 2013)	Preference, Security, Accessibility, Learning Style, Learning Task, Goals, Skills, Interest, Diploma, Identification, Affiliation, Personal Information, Historic, Objective, Competence, Activity, Pre-required Knowledge, Formative Evaluation	18
20	(Lamia & Tayeb 2013)	Level Knowledge, Demographic Information, Gender, Age, Language, Culture, Preference, Navigation, Goal, Level Knowledge, Ability, Learning Style, Thinking Style, Difficulty Level, Media Format	15
21	(Hafidi & Bensebaa 2013)	Personal Data, Knowledge Level, Preferences, Traces, Prior Knowledge, Learning Activity, Assessment, Learning Performance, Propagation, Learning Success Rate, Skill Level, Achievement, Gap Control, Potential Control, Learning Success Rate, Learning Progression, Level of Competence	17
22	(Marković et al. 2013)	User Behaviour, Knowledge State, School, Student Profile, Learning Style, Cognitive Style, Learning Path, Learning Time, Learning History, Knowledge Level, Media, Test Result, Psychological Characteristic, Activity, Learning Process, Achieved Results	16
23	(Legaspi et al. 2008)	Level of Cognitive, Proficiency Level, Effort, Response to Hint, Number of Mistakes, Problem Characteristic, Difficulty Level, Type of Operand, Utility, Number of Subskill Tested, Result Information, Level of Interactivity, Maximum Hint, Current Hint, Amount of time, Task, Age, Gender, Trait	19
Total			**428**

4.1.4 Step 4: Selection of the general concepts that are used in the models

In this step, the list of concepts derived from Table 4.2 was analysed and refined. The first step in selecting a candidate concept was filtering the number of occurrences and generality of the name. The purpose of filtering based on occurrence is to find the concepts with a higher degree of confidence. This process is part of the methodology proposed by (Beydoun, Low et al. 2009). The same concept in each model was counted, then arranged from the highest to the lowest occurrence. If the concept appeared in more than three models, it was selected as a candidate for inclusion in the model list. Meanwhile, the concept that appeared in less than three models was discharged from the process. If two or more concepts had a similar meaning, then these concepts were unified by selecting only one name that had the highest frequency. For example, as shown in Table 4.3, the authors used the concepts: *Difficulty*, *Difficulty Factor*, *Difficulty Level*, and *Level of Difficulty* for the same meaning and purpose. The term "*Difficulty Level*" is designated in the model list as it has the highest frequency and is a more general term. The other concepts are thus excluded from the model list. The concepts that do not directly represent the elements of learners were removed from the list. For example, concepts *Utility* and *Current hint* were removed as they only appear once in a specific model. In case an important concept was found but less than twice, this concept might be too specific to the proposed model. Thus, a process to harmonize and fit the concepts between the candidates is required. This activity will be conducted in the next Step 5.

Table 4.3 Example of selection of general concept

No	General Concept	Concepts	Frequency	Generality
1	Background	Background	4	1
		Background Knowledge	2	1
		Background Domain	1	0
2	Bandwidth	Bandwidth	2	1
		Connectivity	1	1
3	Cognitive Style	Cognitive	1	1
		Cognitive Ability	1	1
		Cognitive Capacity	1	0
		Cognitive Skills	1	1
		Cognitive Processing	1	0
		Cognitive Purpose Indicator	1	0
		Cognitive State	2	1
		Cognitive Style	4	1
		Cognitive Task	1	0
		Level of Cognitive	1	1
		Visual and Cognitive	1	0
4	Competence	Competence	2	1
		Competency	1	1
		Level of Competence	1	1
5	Content	Content	1	1
		Content Experience	1	0
		Content Identification	1	0
		Course Content	1	1
6	Course	Course	1	1
		Course Module	1	1
		Grade of Course	1	0
7	Device	Device	2	1
		Display	1	0
		Interface	2	0
		Learning Device	1	1
		Tool	1	1
8	Difficulty Level	Difficulty	1	1
		Difficulty Factor	1	0
		Difficulty Level	4	1
		Level of Difficulty	1	1

4.1.5 Step 5: Classification of the selected concepts from the previous step

In this step, the selected concepts were collected into a classification according to their function in the learner model. As stated in Step 2, the classification is selected from the literature review based on the general taxonomy of the learner model on AES. The selected concepts were grouped into three classifications: (i) Personal user characteristic; (ii) Previously acquired knowledge and skills; and (iii) System-related user characteristic. Personal user characteristic comprises general user information, such as age, gender, or individual traits. Previously acquired knowledge and skill encompasses prior experience, previously acquired skills, and background knowledge. Lastly, system-related user characteristic includes all concepts that are changeable as related to the particular system. Classification into this category is presented in Table 4.4.

Table 4.4 List of proposed concept organised into three classifications

Classification	Concepts
Personal user characteristic	Language, Age, Cognitive Ability, Competence, Education Status, Emotion, Occupation, Gender, Learning Style, Personal Information, Personality
Previously acquired knowledge and skills	Course, Instructional Plan, Knowledge Level, Background, Learning Experience, Activity, Skills, Learning Performance, Objective, Concept, Prior Knowledge
System-related user characteristic	Bandwidth, Device, Preference, Goal, Task, Teaching History, Time, Topic, Interest, Difficulty, Location, Motivation, Interactivity, Format

4.1.6 Step 6: Validation of the initial learner model

This validation set out to measure the quality of the learner model by investigating the generality, expressiveness and completeness of the proposed learner model. The purpose of the validation here is to ensure that the model contains adequate concepts to represent the purpose model. For this purpose, the validation process follows the frequency-based selection by De Kok (2010), as validation techniques. This method is performed by comparing each candidate concept with the models in Validation Set 1. Then, the frequency of appearance is counted and the concept with a low score is eliminated. The comparison against other 10 models is to identify wether any missing concepts to the initial version. Besides, it will also ensure the initial model covered broad coverage. Table 4.5 illustrates the comparison of candidate concepts with other models.

Table 4.5 List of candidate concepts comparing with models in Validation Set 1

No	Concepts	1	2	3	4	5	6	7	8	9	10	Frequency
	Personal user characteristic											
1	Age											0
2	Cognitive Style	x		x	x						x	4
3	Competency					x			x			2
4	Experience					x		x		x		3
5	Gender											0
6	Language											0
7	Learning Style	x		x		x	x				x	5
8	Qualification								x			1
9	Personal Information											0
10	Personality		x	x	x					x		4
	Previously acquired knowledge and skills											
11	Background	x		x		x		x		x		5
12	Content											0
13	Course	x		x		x						3
14	Domain Concept		x	x	x						x	4
15	Instructional Plan			x								1
16	Knowledge Level	x	x	x		x	x	x	x	x	x	9
17	Learning Activity	x	x				x		x			4
18	Learning Performance			x				x				2
19	Prior Knowledge	x	x	x		x	x	x	x			7
20	Skills		x	x	x				x	x		5
21	Topic											0
	System related user characteristic											
22	Bandwidth										x	1
23	Device					x				x		2
24	Difficulty Level		x	x	x						x	4
25	Emotion		x							x		2
26	Goal		x				x	x		x		4
27	Interactivity				x					x	x	3
28	Interest							x		x		2
29	Learning Motivation		x							x		2
30	Location									x		1
31	Objective						x			x		2
32	Preference		x	x		x	x	x			x	7
33	Presentation	x			x	x			x		x	5
34	Time	x		x	x				x		x	5

Furthermore, frequency-based validation uses an indicator called the *Degree of Confidence* (DoC). DoC value is used to measure the probability that a concept is used in a randomly chosen model that is included in Validation Set 1. DoC can be formulated as follows:

$$Degree\ of\ Confidence\ (DoC) = \frac{Frequency\ of\ Concept}{Total\ Model\ of\ Set\ 1} \times 100\%$$

The value of DoC can be categorised into the following five levels of importance:

- Very Strong (70-100%).

- Strong (50-69%).
- Moderate (30-49%).
- Mild (11-29%).
- Very Mild (0-10%).

Very Strong means that the selected concept has appeared many times in Validation Set 1 models. For example, the concepts *Knowledge Level* has a very strong concept with 90% DoC value. On the other hand, Very Mild has the least appearance of the concepts from zero to one. The result of the DoC classification with the selected concepts is illustrated in Table 4.6.

Table 4.6 Degree of Confidence for Learner Model

DoC Classification	Learner Model Concepts
70-100% (Very Strong)	Knowledge Level, Prior Knowledge, Preference
50-69% (Strong)	Learning Style, Background, Skills, Presentation, Time
30-49% (Moderate)	Cognitive Style, Personality, Domain Concept, Learning Activity, Difficulty Level, Goal, Experience, Course, Interactivity
11-29% (Mild)	Competence, Learning Performance, Device, Emotion, Interest, Learning Motivation, Objective
0-10% (Very Mild)	Qualification, Instructional Plan, Bandwidth, Location, Age(✗), Gender(✗), Language(✗), Personal Information(✗), Content(✗), Topic(✗)

(✗)=Delete the concept

The purpose of this validation is to refine the final decision on whether a concept will be included in the model or not. In this work, the concept that had zero DoC value was deleted from the list of the learner models. However, some concepts that still had a DoC value, even in the Very Mild category with low value, were still included in the learner model. For instance, *Qualification, Instructional Plan, Bandwidth* and *Location* are still in the model because they have 10% DoC value. On the other hand, the concepts *Age, Language, Gender, Personal Information, Content,* and *Topic* were deleted from the learner model because of zero DoC value. Table 4.7 shows the final list of the learner model organised into the three components of the learner model.

Table 4.7 Final concept of learner model

Components	Concepts
Personal user characteristic	Cognitive Style, Competency, Experience, Learning Style, Qualification, Personality
Previously acquired knowledge and skills	Background, Course, Domain Concept, Instructional Plan, Knowledge Level, Learning Activity, Learning Performance, Prior Knowledge, Skill
System-related user characteristic	Bandwidth, Device, Difficulty Level, Emotion, Goal, Objective, Interactivity, Interest, Learning Motivation, Location, Preference, Presentation, Time

4.2 Proposed List of Concept and Definition

This section describes the concepts of learner model in detail. First, the main concepts of each component in the learning model are presented. This is followed by the description of the concept and the illustration of its value during the implementation. The lists of the concept were organised in disorder. Even though some concept might have a different degree of importance when implementing in the domain. However, to determine the degree of importance will need more information related to the implemented domain. Without understanding the implemented domain, it would be impractical to compare the importance of the concepts.

4.2.1 Personal user characteristics

Personal user characteristics encompass information regarding the general user and individual traits. Information related to the general user, such as name, age, competency, experiences, qualification, could be captured directly from the user's feedback. This mechanism differs from individual trait. Individual traits, including cognitive style, learning style, and personality, usually require separate testing and interpretation of the results.

Cognitive Style. The term cognitive style can be described as the way students think, observe, and remember information. It encompasses the working memory, control and speed of processing (Germanakos et al. 2008). Cognitive style, along with learning style, have become the essential concepts in adaptive learning. Researchers have identified that the cognitive processing of the human is often related to age, exercise, and experience (Granić & Nakić 2010). According to the Wechsler Adult Intelligent Scale (WAIS), the cognitive style could be classified into six broad types: *information, comprehension, vocabulary, similarities, object assembly, and picture arrangement* (Peña Ayala 2010). Another general classification of cognitive style is field dependence and field independence (Jun-Ming et al. 2006).

Competency. Competency indicates the qualification of a learner in a domain. Although this is not compulsory for the user model, competency might be included as the portfolio of the learner (Jun-Ming et al. 2006). Competency could represent that a student possesses the skills and knowledge required in a particular field. A student with technologically related certification, for example, demonstrates the competency and knowledge that can be used to generate or personalise content learning (Martins et al. 2008).

Experience. Experience represents the prior record associated with the learner's previous practice outside the core domain of the system. Some authors (Colace et al. 2016; Medina-Medina, Molina-Ortiz & García-Cabrera 2011) define experience as the user's general knowledge about the domain. However, we include the knowledge inside the domain in the learning history, as suggested by (Dogan & Dikbıyık 2016; Jeremić, Jovanović & Gašević 2012). Furthermore, experience represents a range of backgrounds that might be used in AES, including job, profession, responsibilities, or working experiences in related areas.

Learning Style. Learning style is the way a student learns or prefers to learn. In fact, every student has different methods or strategies which they prefer when studying. Many studies have been done to investigate the specific learning style of students to improve the learning process (Brown et al. 2009; Feldman, Monteserin & Amandi 2015). The selection of the appropriate learning style was derived from educational and psychological field. In e-learning environments, the common learning styles adopted for adaptive learning are Riding's Cognitive Style Analysis (Germanakos et al. 2008), Felder/Silverman Index of Learning Styles (Özpolat & Akar 2009; Yang et al. 2014), Myers–Briggs type indicator (Kim, Lee & Ryu 2013; Rastegarmoghadam & Ziarati 2016), and Kolb's Model (Papadimitriou, Grigoriadou & Gyftodimos 2009).

Qualification. The qualification indicates the learner's level of education. IMS uses education context to describe the environment where learning activities take place, such as University, School, Training, and others (IEEE 2002). In AES, the qualification is used not only to represent the learning resource but also to represent attributes of learners. Qualification can also be used to generate and personalise the content that is matched with the education context (Jun-Ming et al. 2006; Sangineto et al. 2008).

Personality. Personality reflects a record that stores relevant learning attitude of a learner. According to (Kim, Lee & Ryu 2013), learner's experience in e-learning will be affected significantly if the instruction style matches the learner's personality. That is why the modelling of personality is often closely tied to the preference for learning resource presentation. Some studies proposed different strategies of personality types, such as Keirsey's Temperament Theory, Five-Factor Model, and MBTI. Daughenbaugh et al. (2002) demonstrated four personality types that are highly relevant to learning: *Rational* (intuitive thinking), *Idealist* (intuitive feeling), *Artisan* (sensory perception), and *Guardian* (sensory judgment). The Five-Factor Model as suggested by McCrae & John (1992) also has different effects on learning. This personality consists of five dimensions: *openness, conscientiousness, extraversion, agreeableness,* and *neuroticism*. The MBTI model by Harrington & Loffredo (2010) provides personality preferences along four bipolar psychological dimensions: *Introvert–Extravert, Sensing-Intuition, Thinking-Feeling,* and *Judging-Perceiving*.

4.2.2 Previously acquired knowledge and skills

This category of learner model comprises information related to previously acquired knowledge, such as background of the learner, domain concept, prior experience and knowledge, previously acquired skills and learning activity.

Background. In broad terms, background refers to the students' previous knowledge in some particular areas, usually outside the domain of the concept, but still relevant to the body of knowledge. For example, the background for students in the field of Computer Science or Information Technology might contain information related to knowledge in programming, system and development, networking, and operating systems. The other fields will contain the different information of the background.

Domain Concept. A domain concept consists of a topic of the domain. The domain concept can also be associated with learning objects in e-learning. The relationship among domain concepts of the course is known as the learning path. Information related to domain concept is crucial for adaptation. This allows the system to generate and predict the content for the student based on the information in the domain concept (Sangineto et al. 2008). The information in the domain concept is also needed for determining the adaptation strategy, which is generally defined in the instructional model or adaptation model.

Instructional Plan. The instructional plan refers to a specific kind of strategy that is used to provide an effective design of learning object to reach a more successful learning outcome. The instructional plan focuses on current learning states, needs, and learning outcomes of students. Also, the instructional plan is a developing functional learning system based on a systematic approach that meets the requirements of a specific target group.

Knowledge Level. Knowledge level represents an element record that reflects the expertise level of knowledge about a topic given. Knowledge level could represent the *acquired knowledge* or *required knowledge.* The acquired knowledge is the knowledge that the student acquires at a given level. This is also given to the student after they pass a chapter or a test, such as in (Limongelli et al. 2009). The required knowledge describes the necessary knowledge that students should have before studying a material (El Bouhdidi, Ghailani & Fennan 2013). Knowledge level is essential in adaptive learning. A system can adapt the level of student and decide what the next stage of the learning process might be by accessing the knowledge of the learners. This way of adaptation is promoted by (Colace et al. 2016; Hafidi & Bensebaa 2013; Medina-Medina, Molina-Ortiz & García-Cabrera 2011; Papanikolaou 2015). Knowledge level can also be reflected using performance level indicators (Verginis et al. 2011), automatic system score (Chrysafiadi & Virvou 2012), or question items level, such as *Well Learned, Learned, Poor,* and *Very Poor* (Dogan & Dikbıyık 2016).

Learning Activity. Learning activity is defined as any activity that the learner does while progressing through a unit of instruction in the course. Several types of learning activities exist according to cognitive or learning style. For instance, Felder-Silverman exemplifies learning activity as case studies, brainstorming, discussion, problem solving, experiments, observing, mind maps, group discussion, case studies, reading, storytelling, interviews, listening to lectures, questionnaires, and consulting references (Ouf et al. 2017).

Learning Performance. Learning performance aims to measure the achievements and effectiveness of different kinds of learning activity. Learning performance can be determined by several methods. First is using learning goal or task accomplishment rates. Second, the general measuring score of pre or post-test after completing a lesson. Then, academic performance by grading the result of pre and post-test. An effective assessment is a vital way to measure learners' performance and accomplishment of the learning outcomes. That is what learners are expected to know, understand, and be able to do in order to be successful in a domain concept.

Prior Knowledge. Prior knowledge is a way to represent the accumulation of knowledge and learning that each learner brings to a particular course. Prior knowledge is aimed at keeping track of everything that the student has done during the learning process. It is a form of reflection of the performance or success/failure rate of the student. Prior knowledge data can be obtained from the result of the test (Chen, Chang & Wang 2008; Jeremić, Jovanović & Gašević 2012), history of last visited learning resources (Colace et al. 2016; Medina-Medina, Molina-Ortiz & García-Cabrera 2011), or time spent on completing a chapter (Dogan & Dikbıyık 2016).

Skill. Skill represents the ability of a learner to perform well in a particular area. The skill of learner is often determined to personalise learning content based on the difficulty level. Skill might be used to represent a general ability in learning or just in a specific domain. For example, (Jeremić, Jovanović & Gašević 2012) uses skill to measure the degree of mastery in a specific programming language. The values range from *bad* (no skill) to *expert* (high skill). On the other hand, skill also can reflect the general ability of the learner, such as mathematical skill, analytical, problem solving, reasoning and logical thinking (Brusilovsky & Millán 2007; Romero & Ventura 2010).

4.2.3 System-related characteristic

Included in this category is the user characteristic that depends on the system. Therefore, this category encompasses all information related to the interaction between user, system, and the environment. Other inclusions in this category are device, bandwidth, difficulty level, emotion, goal, objective, interactivity, interest, learning motivation, location, preference, presentation, and time.

Bandwidth. Bandwidth provides information related to the network or internet connectivity. Some of the internet connectivity elements include: RFID, IrDA, Bluetooth, WiFi, WiMax, UMTS, 4G, or Satellite (Economides 2009). Moreover, bandwidth elements may include static and dynamic properties. The static properties may be a range of elements to represent the bandwidth values, such as low, medium, high or minimum, maximum (Verbert et al. 2012); while the dynamic properties display the value of the actual speed or capacity of the network (Benlamri & Zhang 2014). Knowing the network connection gives insight into the e-learning system, to provide contents adaptation related to media presentation or size of resources that can be retrieved into student's device.

Device. Device contains information related to the type of equipment that is used for learning. The information that is attached to a device contains technical elements, such as screen size, connectivity, central processing unit, interface, input/output, memory and storage capacity, and battery lifetime (Germanakos et al. 2008). From these characteristics, we could refer to the hardware that the learner uses, such as PC, laptop, PDA, Smartphone, or Tablet (Economides 2009). The device information is important to determine the media presentation or learning resource that would be adaptable to the student.

Difficulty Level. Difficulty level indicates the difference in the task that is hard to accomplish or understood by students. As stated by Chen (2008), the difficulty level is relevant to determine the learners' abilities because inappropriate content can result in disorientation during learning. In particular, he proposed the dynamic parameter of difficulty level in different stages of learning to deliver appropriate learning content. In contrast to this method, some studies used the static parameter to define the difficulty level. For instance, Chakraborty, Roy & Basu (2010) defined three values of *easy, normal,* and *difficult*, as the difficulty parameter. Jeong (2013) proposed a five-step difficulty level, which consists of *Excellent, Advanced, Intermediate, Intermediate Low,* and *Beginner*. Another static parameter as suggested by (Dogan & Dikbıyık 2016;

Jovanović, Gašević & Devedžić 2006) defined difficulty level by using the scale from 0 (easy) to 1 (difficult).

Emotion. Emotion is described as a diffuse affective state that consists of a subjective feeling change without apparent cause (Reategui, Boff & Campbell 2008). Different from mood, which tends to appear when the things are not going well according to their expectation, emotion tends to come from a known cause. For example, learners exhibit a happy emotion when they understand a topic or pass the test. Nevertheless, emotion is often combined and used interchangeably with mood to represent the affective states of the student (Hernández, Sucar & Arroyo-Figueroa 2013).

Goal. Goal is the description of an objective that the learner tries to achieve. The goal in AES is simply the answer to the question *"What does the student want to achieve?"* The goal can be modelled with possible learner goals or tasks that the system can recognise (Brusilovsky & Millán 2007). Typically, a learner has to select one of the pre-defined goals. In (Mahnane, Laskri & Trigano 2013), goal's information is stored to indicate the particular topic in the system that the learner wants to visit. Some systems can capture goals from users and models as a probabilistic overlay (Brusilovsky & Millán 2007). Lamia & Tayeb (2013) suggested two classifications of goal, that is the long-term goal and short-term goal. The long-term goal is the lifelong study plan and typically permanent in the course. In contrast, the short-term goal provides the learner the opportunity to solve a certain problem, such as passing an assignment or doing simulation. Goals can be modelled through navigation or monitoring dashboards. Learning goals are usually set at the beginning of the course, so that the system can provide information related to the goals that the learner wants to achieve. Then, the system might provide the needed resources to achieve the goals or remind the learners of their expected learning level (Chen, Chang & Wang 2008).

Objective. The objective defines a classification of educational objectives that are used by learners to formulate their intentions. An objective in a subject can be measured by a degree of control that learners seek to achieve (El Bouhdidi, Ghailani & Fennan 2013). Many objectives of study were derived from Bloom's taxonomy. According to Bloom's taxonomy, the objective of study can be classified into six categories: knowledge, understanding, application, analysis, synthesis and evaluation.

Interactivity. Interactivity level is described as the degree to which the student can influence the learning resource. Interactivity level promotes opportunities to interact with learning resource in different ways. Moreover, interactivity level indicates the degree to which the learning resource can respond to the actions and input of the user (Friesen, Fisher & Roberts 2004). It has a value space range from very low to very high. The learning activity in very low interactivity level relates to passive activities with less response from user, such as reading an essay or answering test questions. Whereas the very high interactivity level value often relates to active response from user, such as a 3D simulation environment that needs the user to do a series of steps.

Interest. Interest is an indication of a learner being more attracted to a particular learning topic than others. This allows adaptive systems to make lessons that match with the learner's interest. To represent user interest, some authors suggest the keywords-level model and concept-level model (Brusilovsky & Millán 2007). The keyword-level model uses weighed vector of keywords to filter and retrieve the learning materials that are relevant to the keyword. In contrast to this approach, a concept-level model uses a weighted overlay of concept-level domain model to represent user interests.

Learning Motivation. Learning motivation is defined as the desire or the willingness to make an effort towards a specific learning goal (Popescu 2009). Learning motivation is often classified as a system-related characteristic because it intensely depends on the learning goal. The learning goal itself is system-dependent and can be affected by the presentation of learning contents.

Location. Location is the representation of the place or position where the learner can be located. The location could be located automatically using position measurement from GPS or cellular ID (Economides 2009). However, most of them are usually selected manually by users according to the type of settings like a classroom, library, laboratory, home, outdoor, train, or bus (Lee & Lee 2012). Some works allow learners to create initial settings which consist of context attributes, such as location, time and date (Müller & Kohlhase 2009; Pernas et al. 2012).

Preference. Preferences are the learning features that relate to likes and dislikes, in term of interfaces, services, or resources. Preferences such as font type, size, and other parameters are associated with the interface (Chen, Chang & Wang 2008; Lamia & Tayeb 2013). The variables relate to device communication, hardware, and software which are associated with the service (Abech et al. 2016; Benlamri & Zhang 2014).

Furthermore, the preferred type of courses or instructional model can be associated with resources (Gómez et al. 2014). Different methods have been proposed to gather learning preferences from system. Ouf et al. (2017) suggest using questionnaires for collecting learner preferences. This method provides real-time information without any need for further processing. Moreover, this method is similar to Lamia & Tayeb (2013), which provides checklists for users to select preferred interface elements. After the preferences are determined, the system applies adaptation in the new contexts. The preference can also be defined as an ordered list of values, which indicate their likes and dislikes (Chen, Chang & Wang 2008).

Presentation. Presentation is described as the technical datatype or media type of the learning resource that is intended for presentation to the user. The information on presentation can be used to identify the hardware and software needed to access the resources. The type of presentation may take several multimedia forms, such as text, picture, video, audio, application, animation, or hypertext (Dogan & Dikbıyık 2016; Jun-Ming et al. 2006; Lamia & Tayeb 2013; Peña Ayala 2010).

Time. In adaptive e-learning, time could represent the plan of activity (Müller & Kohlhase 2009; Shi et al. 2013), duration spent on studying (Jeremić, Jovanović & Gašević 2012; Limongelli et al. 2009), or indicate the date of learning (Chen, Chang & Wang 2008; Verginis et al. 2011). Similar with location, time context is often used in conjunction with another context, either schedule, place, or motivation (Verbert et al. 2012).

4.3 Chapter Summary

This chapter has discussed the development of a learner model for AES. The first section presented the overview and taxonomy of the learner model from a historical perspective. Then, we synthesised the learner model by investigating notable literature in journals from 2005-2017. The learner models in each study are extracted and reconciled according to three components: user profile, user knowledge and system related characteristics.

The next chapter focusses on the construction of the intelligent module that will guide the adaptation of the learning process. The chapter discusses the overview of technological methodology to support adaptive e-learning systems throughout incremental knowledge acquisition ripple down rules (which was also described in Chapter 2). The development methodology is presented along with the dynamic evaluation for knowledge-based systems. The dynamic evaluation allows a KBS to be evaluated quickly during the knowledge acquisition process. Through ripple-down rules, the knowledge base will be constructed incrementally. Further, the extension monitoring module will analyse and evaluate the adequacy of the knowledge base. The methodology will be deployed applying the model synthesized in this chapter.

5 DYNAMIC EVALUATION METHOD FOR RDR KNOWLEDGE-BASED SYSTEM

Chapter 4 has identified a learner model to enable the representation of requirements of adaptive learning. The proposed model represents properties of the user in AES. This also provides key attributes for test cases to validate the architecture and the knowledge monitoring approach. These two will be the focus of this chapter which essentially presents the design of the KBS with incremental knowledge acquisition RDR and the extension monitoring module for analysing the adequacy of the knowledge base. The incremental development itself has these two requirements: First, the system should guide the expert knowledge in focussing on the articulation and the structuring of the knowledge as they are prompted with a new context. To ensure that the knowledge addition process is tractable, rules will only be added rather deleted or modified. The insertion points will be automatically identified from errors that are found during the knowledge acquisition process. The role of the expert will be restricted to creating new rules. This mechanism is essentially the strategy of RDR knowledge base maintenance. The second requirement is that the system should facilitate the monitoring of the knowledge acquisition process. This is a key contribution of this thesis to the knowledge acquisition community. A monitoring module is added to calculate statistical analysis on the fly during the knowledge acquisition process. The monitoring module will present the performance of the KB for further analysis as the knowledge base is developed. As this is a key conceptual contribution of the thesis, the chapter is heavily concerned with the readiness of the knowledge base. It describes how the dynamic evaluation framework can be applied to RDR KB development.

The remainder of this chapter is organised as follows. Section 5.1 presents the philosophical background of the validation process of knowledge-based systems. The proposed dynamic evaluation methodology is then discussed in Section 5.2. The architecture design of a prototype for AES is introduced in Section 5.3. Finally, the summary of work in this chapter is presented in Section 5.4.

5.1 The Dynamic Evaluation (DE) Framework

Our framework is based on the dynamic evaluation approach as introduced by (Beydoun & Hoffmann 2013). The dynamic evaluation focuses on validating the KBS without separating the development and testing phases. The DE relies on the availability of the test cases and works well with the manually constructed knowledge base, such as RDR. Since RDR uses the incremental knowledge acquisition process to build rules, the DE works by monitoring each case entered into the system. Statistical analysis is utilised to estimate the error level of the developed KB with a certain degree of confidence. In particular, the estimation of binomial proportion is applied to determine the probability of correct cases that fall into the context of a particular rule in a sequence of test cases. Furthermore, the monitoring process of KBS is presented from the beginning of the inference. That is when the first rules are created until the expected performance has been reached. Thus, this approach is classified as '*dynamic*' by nature because the stopping validation will differ between conditions of the KB. It is determined 'on the fly' as the knowledge base is developed.

There are two reasons for using the dynamic evaluation approach in this study. First, the knowledge base is never complete due to changes and new discoveries being made at any time (Ramadan et al. 1998). There would always be new rules that would have to be added. Thus, we believe that knowledge base should include sufficient knowledge and all the necessary knowledge domains to answer a given case. Second, the process of developing a knowledge base often requires a separate testing phase from the system's development because of the difference in nature. The dynamic evaluation approach merges the design, implementation, and maintenance phases to ensure that the KB already has enough knowledge to deliver and to be ready to use immediately. Dynamic evaluation advocates statistical monitoring to gauge the expertise quality and suitability of domain of the knowledge base.

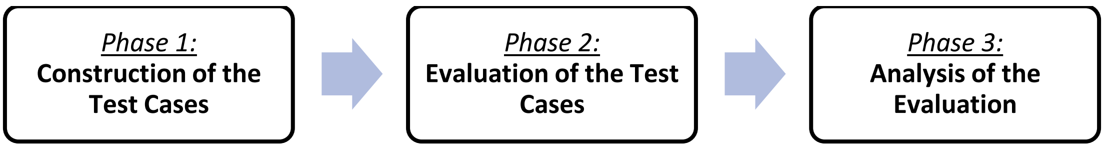


Figure 5.1 Dynamic Evaluation Framework

As shown in Figure 5.1 above, we proposed a three-phase strategy for DE framework with the initial step as modelling the domain of the knowledge. The brief description of the phases are as follows:

1) The first phase incorporates data stream creation for knowledge evaluation. The knowledge representation is vital in the evaluation process of the KB. Test case data provided should represent the domain to be tested. Therefore, the test cases creation should be selected based on the criteria that represent the real cases that will be faced by the system.

2) The second phase of validation concerns with testing every input of the test case, created from the previous phase of the system. If the test case is proven to be possible in the real world, the KB should work correctly and guarantee that the result is valid.

3) The third phase evaluates the testing result by monitoring the statistical data of knowledge acquisition with the iterative testing phase. Statistical monitoring is built to estimate the coverage, error, and predictivity for the KBS being validated. Furthermore, the results of this validation method are monitored and measured by analysing the effectiveness of the KB. The second and third phases will be executed iteratively until a stopping condition is met. The stopping condition is determined by the accuracy level sought. The final decision of this process will conclude the results and stop performing validation. Summary of the validation would indicate when the knowledge acquisition must be terminated with the decision being the acceptance or rejection of the knowledge base.

The next subsections describe further details of purposes and specific tasks or activities of each phase that should be followed.

5.1.1 Phase 1: Construction of the Test Cases

The main aim of Phase 1 is to provide a set of test cases for evaluation. A test case is a set of test data combinations that consists of inputs, executions conditions, and expected outputs (IEEE 2005). Test case creation is difficult and expensive in practice. Hence it is necessary to set the selection criteria vigorously to minimise the refinement.

During the evaluation, it is not possible to perform thorough testing, where all input parameters will be tested. Therefore, the availability of the test cases is essential in dynamic evaluation approach.

5.1.1.1 Test Case Extraction and Format

The first step towards formalising the test case construction is to extract them from the domain knowledge and present them according to the required format. The format of the test cases in DE has the following five variables:

1. Test Case ID: the number of the test case, starting from one and increasing by one every test case.
2. Input Variables: the parameters and values of input of the test case.
3. Expected Output: the expected output defined by the expert for every test case.
4. System Output: the actual output from the system after executing the test case.
5. Description: A comment or description of the test case.

Each state of the input variable has an answer (expected output) defined by the expert. These answers will be compared to the actual output from the system. The expert might have noted some judgement or further comments in the description. The input variables are generated from the domain model, which represents the parameters and each possible value. As an example, Colace, De Santo & Greco (2014) suggested a user model of adaptive learning with five characteristics, as shown in the following Table 5.1.

Table 5.1 Example of Parameters and Values

Parameters	Values
Type of media	Video, Audio, Text
Interactivity level	Perception, Input, Processing, Understanding
Bandwidth	Low, High
Difficulty level	Easy, Medium, Difficult
Time of study	10m, 20m, 30m

The above parameters and values can be extracted into input variables. These attributes will be the primary input variables, which consists of attribute names and their value, which describes a learner. As illustrated in Table 5.2, each of the input variables has an expected answer. This is the lesson name defined by an expert that should be given to the learner in a particular situation. When evaluating the test cases, each input gets an actual answer given by the system. So, every test case reflects a learner and the knowledge that should be allocated in such a situation. The procedure to generate test cases will be discussed in the following sections.

Table 5.2 An example of test case extraction from ALS

Test Case ID	Input Variables	Expected Output	System Output	Description
1	Type of media: Video; Interactivity level: Perception; Bandwidth: High; Difficulty level: Medium; Time of study: 10m	Lesson 1	Lesson 2	Rule created
2	Type of media: Video; Interactivity level: Input; Bandwidth: Low; Difficulty level: Easy; Time of study: 20m	Lesson 5	Lesson 5	-
3	Type of media: Text; Interactivity level: Understanding; Bandwidth: High; Difficulty level: Medium; Time of study: 30m	Lesson 7	Lesson 6	Rule created

5.1.1.2 Test Case Generation

There are several strategies available for generating test cases. However, DE works by monitoring the knowledge acquisition process, not by evaluating a certain number of test cases. Therefore, DE requires optimum data that represents the correct test case and real coverage of the domain knowledge. To make it practical, the requirements for the test cases are coverage and a minimum number of test cases. For that reason, a *pairwise testing* method is utilised to compromise between requirements. Pairwise testing is an approach to combinatorial testing problems (Czerwonka 2006; Sanchez 2016). This approach includes identifying parameters that define the possible space of test scenarios, then selectively choosing test cases to cover all the pairwise between parameters and their values. Pairwise testing will cover all possible pairs of parameter values by at least one test case (Kuhn, Kacker & Lei 2013). The goal of pairwise testing is to seek for the minimum set of a valid test cases with a significant test coverage. It will provide excellent coverage, and the number of test cases will remain manageable.

In this process, each pair (or t-wise) interaction will occur at least in one test case, but it may occur in more than one test case. The pairwise process creates a tabular representation of the system's parameters. There should be at least one test case per row, where each row represents the combinations of values for each parameter. For example, given three parameters from data in Table 5.2: Type of media (A), Interactivity level (B), and Bandwidth (C). The structure of pairwise generation is set up to slots AB, AC, and BC. Each of these slots corresponds to possible value combinations, such as for AB={Video-Perception, Video-Input, Video-Processing, ... , Text-Understanding} or AC={Video-Low, Video-High, Audio-Low, ... , Text-High}.

Each slot can be marked as covered, uncovered, or excluded. Covered means the algorithm produces a test case that satisfies a specific combination, and vice versa, for the uncovered. Excluded means that the slots are removed from combination to be covered. The generation algorithm is a greedy heuristic. It builds one test case and terminates if no more covered slots found. This process ensures that the result will have more coverage within fewer test cases. It also reduces the combinatorial explosion that can occur when attempting to test a system with many input options, as is the case in this research. Even compared with the exhaustive test data sets, pairwise testing generates a smaller test case. Therefore, pairwise testing is a suitable approach when the processing involves interaction between parameters (Kuhn, Kacker & Lei 2013).

In this research, pairwise testing is applied with an excel-based test generation tool called PictMaster (Software 2016). As the name implies, the tool overlays PICT with an excel interface (Czerwonka 2006). PICT is a free application that generates combination testing based on command prompt. PictMaster is provided as a free software and as an open-source tool. PictMaster requires several steps to generate the test case, which are described as follows:

Step 1: Input Parameters and Value Hierarchy

The first step requests the user to enter parameters and value hierarchy. Parameters contain names of attributes in a data model. Each parameter consists of a finite number of possible values. The value hierarchy is separated by commas (,) and can be modelled as aliasing (by symbol |). Aliasing is a way of specifying multiple names for a single value. This mechanism treats multiple values as one entity. It works by rotating the value's name among the case and reducing the combinatorial complexity of the model. Figure 5.2 illustrates the parameters and value hierarchy from the data in Table 5.1. This example consists of 5 parameters and 15 values. However, the Video and Audio are aliased and counted as one entity. So, Video and Audio will be rotated among the test cases.

Parameters	value hierarchy
Type of media	Video | Audio, Text
Interactivity level	Perception, Input, Processing, Understanding
Bandwidth	Low, High
Difficulty level	Easy, Medium, Difficult
Time of study	10m, 20m, 30m

© IWATSU System & Software Co., Ltd.

Figure 5.2 Example of parameters and value hierarchy

Step 2: Input Constraints Table

Constraints allow users to limit the domain by specifying the unwanted combination of parameters and values. By this mechanism, the unwanted combinations can be excluded from the result, leaving the required combinations only. As an example, one pair that will occur at least in one test case is the pair of Video (Type of media) and Low (Bandwidth). In reality, this combination is unlikely as the media format Video will need High bandwidth. These violating test cases could not simply be removed from the result as they might cover other possible valid pairs. Instead of losing valid pairs, the pairwise testing method eliminates disallowed combinations through constraint mechanisms. PictMaster uses *If* and *Then* relations to express constraint conditions and targets. *If* constraint condition is specified, *Then* constraint target will be generated. The constraint expression applied for the model is shown in Figure 5.3.

Constraints table			
Parameters	Constraint 1	Constraint 2	Constraint 3
Type of media	Video		
Interactivity level			
Bandwidth	High		
Difficulty level			
Time of study			

Figure 5.3 Constraint Table

Step 3: Configure the Settings

As shown in Figure 5.4, the Settings menu has all information related to the test configurations. In this model, we select the "*Use constraints table*" to apply constraints to the test case. "*Optimize constraint expression*" allows PictMaster to optimise the constraint expressions that are generated from the constraints table. It means that if the system determined that it will take a long time to generate test cases due to the inappropriate constraints, the format of constraints would be optimised automatically to complete the generation in less time. "*Show statistical information*" will display information related to the frequency of generation, a number of test cases, and time to complete the test case generation once the process finishes. "*Show coverage*" displays the combination coverage proportion which has been created during the test case generation. In this model, we use specific 4-way coverage with 90% values to ensure that output test case will cover 90% of the domain. The generation process will be repeated five times to ensure that the output of the test cases has the desired test coverage.

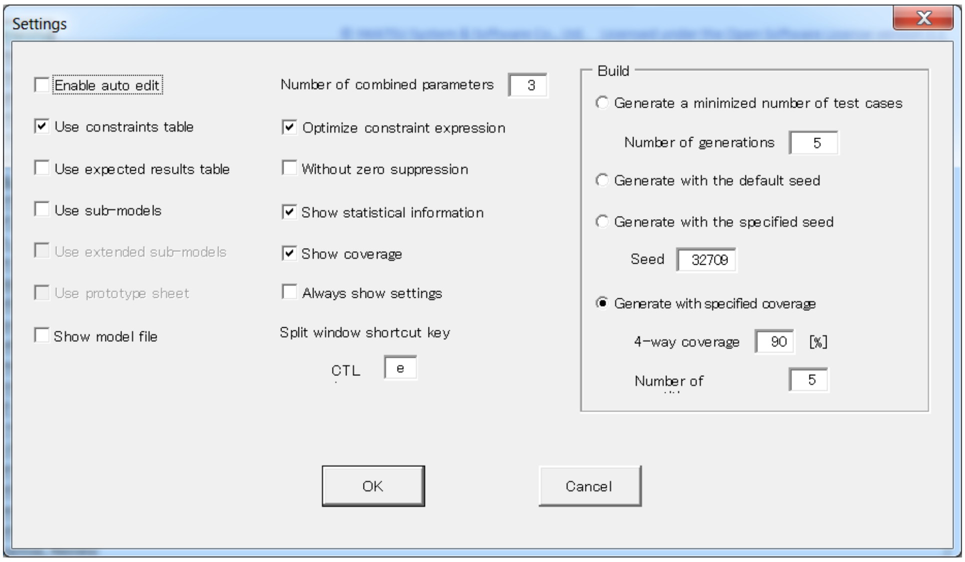


Figure 5.4 Settings Menu

Step 4: Build the Test Cases

By clicking the "Build" button, the tool will start the generation process confirming the indicated parameters, values hierarchy, and constraint conditions. After the process has finished, the system displays statistical Information to the user. As shown in Figure 5.5, the tool produced 71 test cases with five repetitions. This included 100% 3-way coverage and 90.1% 4-way coverage. The overall process took 20 seconds to gather the results.

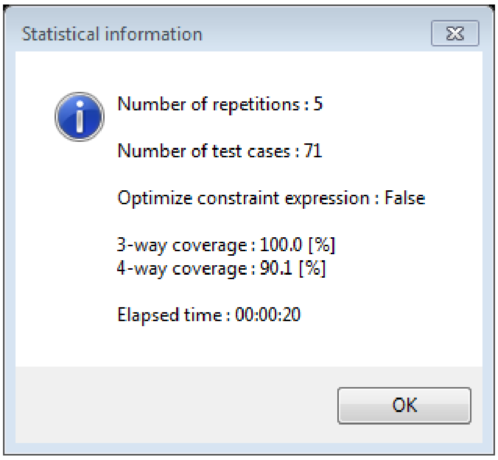


Figure 5.5 Statistical information of the generated input

After we click Ok, the system will show the list of the generated test cases in an excel file. The last step is completing the result according to the format in Table 5.2, which appends three fields: Expected Output, System Output, and Description. The final test cases are shown in Figure 5.6.

Test Case ID	Input Variables					Expected Output	System Output	Description
	Type of media	Interactivity level	Bandwidth	Difficulty level	Time of study			
1	Text	Understanding	High	Medium	20m	Lesson 1		
2	Text	Processing	High	Medium	10m	Lesson 2		
3	Video	Input	High	Medium	10m	Lesson 3		
4	Text	Perception	Low	Medium	30m	Lesson 4		
5	Audio	Understanding	High	Medium	20m	Lesson 5		
6	Text	Processing	Low	Easy	30m	Lesson 6		
7	Text	Processing	Low	Difficult	30m	Lesson 7		
8	Text	Perception	Low	Difficult	10m	Lesson 1		
9	Text	Perception	Low	Easy	10m	Lesson 1		
10	Video	Processing	High	Medium	20m	Lesson 2		
11	Text	Understanding	Low	Difficult	30m	Lesson 3		
12	Text	Input	Low	Medium	20m	Lesson 4		
13	Text	Input	Low	Difficult	10m	Lesson 5		
14	Audio	Understanding	High	Difficult	10m	Lesson 7		
15	Text	Input	High	Easy	20m	Lesson 4		
16	Text	Input	Low	Medium	30m	Lesson 5		
17	Text	Processing	Low	Difficult	20m	Lesson 6		
18	Text	Understanding	Low	Difficult	10m	Lesson 7		
19	Text	Understanding	Low	Easy	30m	Lesson 1		
20	Video	Processing	High	Difficult	30m	Lesson 2		

Figure 5.6 List of the test cases

5.1.2 Phase 2: Evaluation of the Test Cases

Phase 2 focuses on the development of the knowledge base by evaluating the created test cases. In this phase, the information of the learner model captured from the previous phase is gathered and transferred into the further mechanism. Acquiring knowledge often becomes a bottleneck since the task is difficult and time-consuming. However, acquiring knowledge from the presented case into a structured model is crucial during knowledge base construction. This process translates the presentation of procedural knowledge into deep knowledge, that has meaningful knowledge related to the domain. The educational expert is involved in coping with the problem when understanding the principle or relationship between the presented case and the problem domain.

An incremental knowledge acquisition approach using single classification RDR was selected for its reliability in building a knowledge base. Different from other AES that need all adaptation rules to be defined first, our tool does not need to model or define the adaptation rules at once. All rules are built incrementally by extracting conditions from the presented cases when experts find an error. This error correcting process is particularly useful in maintaining the KB as this will keep only valid rules to be stored in the KB. The knowledge acquisition process for constructing a knowledge base is shown in Figure 5.7.

The knowledge acquisition process has the following steps:

1) The first step is to present a learning case to a knowledge base.
2) The inference mechanism begins for the presented case. Then, the system provides a conclusion for the presented case.
3) If the expert agrees with the conclusion, the process finishes.
4) If the expert disagrees with the conclusion, a refinement rule is added to correct that conclusion. The corrected mechanism begins with the expert selecting the expected conclusion. Then, the expert must provide the conditions for the new rule. The system stores the conditions and conclusions as a new rule. The case that causes rule addition is also stored for reference as a cornerstone case.

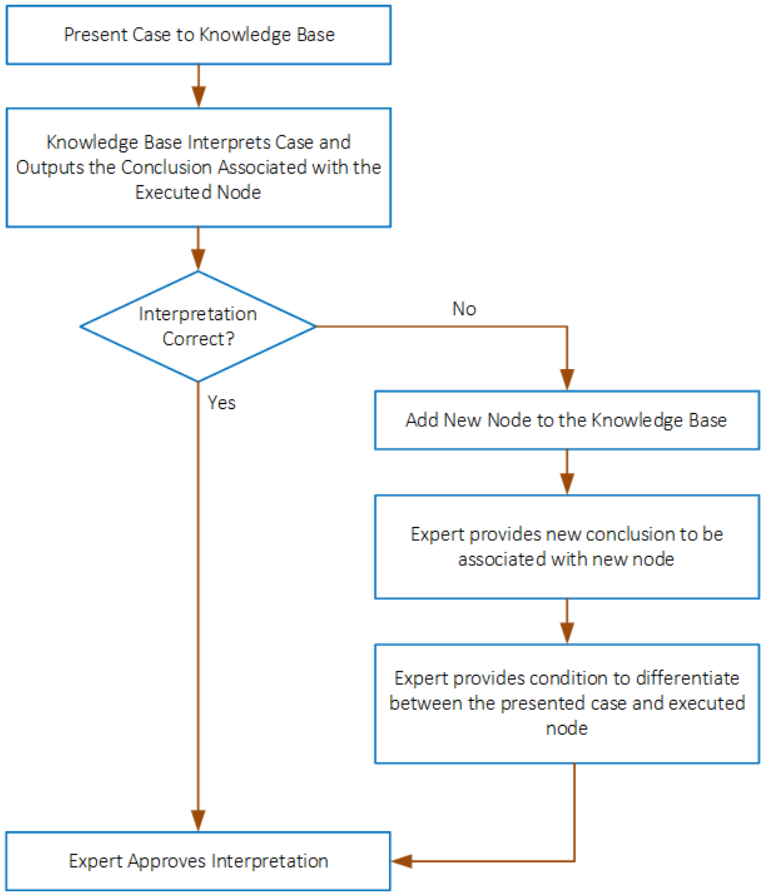


Figure 5.7 Knowledge acquisition process for constructing KB

In RDR, the first inference starts when the first rule is added into the KB. This first rule is called the "*default rule*" and has a default conclusion that will satisfy all cases. Then, when the first case is entered into the KB, it is evaluated against the first rule. The system gives a tentative conclusion to an expert. To illustrate the inference mechanism, a real example of the cases and the process to construct a knowledge base is presented in the following:

1) The process begins with a default rule (R0) in the KB. An RDR structure always has a default rule as a root node, which is always true. The default rule is used to give a solution to any presented condition. When the conclusion from the default rule is not acceptable, the expert needs to provide a new conclusion and refine the condition for the new context. The new rule will be stored as a false link of the first node of the default node. As illustrated in Figure 5.8, the default rule has no condition, which means any attributes will always be true and have the default

answer. The default conclusion for R0 is the introduction chapter: ‘*Message to the Student*’.

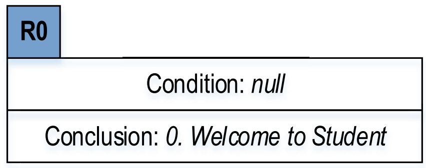


Figure 5.8 An empty RDR Knowledge Base with a default rule

2) Figure 5.9 illustrates four learning cases which are presented to the system. First, Case 1 is evaluated against the KB, which only has a default rule. Because the default rule always gives a conclusion, the ‘*Message to the Student*’ is obtained as the temporary conclusion. If the expert agrees with the conclusion, then the inference process finishes. However, the expected conclusion from the system is different from the actual conclusion. Thus, the expert provides a new conclusion for Case 1 and selects the features for the context of Case 1. For instance, the expert decides that the conclusion is ‘*7.2.1. ICMP*’, because learner has only *15 minutes* to learn in *outdoor* space and has study concept ‘*7.1. IPv4 Network Addresses*’.
   Both conditions and conclusions are stored as a child node of R0. Therefore, the KB has two nodes now, which are R0 and R1 (Figure 5.10). All information in Case 1 is also stored as a cornerstone case for R1 since it causes R1 to be created in the KB. This information is useful for the next inference process.

3) The next case, Case 2, encounters the same process of inference as Case 1. Firstly, Case 2 is evaluated against R0 and the condition is always true. Then, Case 2 follows the exception path and is evaluated against R1. As the rule R1 fires, the condition of Case 2 iss compared to the condition of R1. It turns out that both Case 2 and R1 have a similar condition in ‘*Time: 15 m*’ and ‘*Location: Outdoor*’. Therefore, the system provides the conclusion “*7.2.1. ICMP*” to the expert. The expert agrees with the conclusion and the process is halted.

ATTRIBUTES	Case 1	Case 2	Case 3	Case 4
Background	Programming	Web Technology	Web Technology	Computer Networking
Competence	IT Certification	non-IT Certification	No certification	No certification
Course	Information Technology	Information Technology	Information Technology	Information Technology
Experience	Placements	No experience	Placements	Graduate Certificate/Diploma
Qualification	Diploma	Bachelor	Graduate Certificate/Diploma	Work Experience
Bandwidth	Low	High	High	Low
Device	Smartphone	Tablet	PC	PC
Emotion	Happiness	Pride	Pride	Elation
Learning Motivation	Unmotivated	Average	Average	Average
Location	Outdoor	Outdoor	Library	Home
Time	15 m	15 m	45 m	30 m
Goal	Complete a Lesson	Complete a Lesson	Perform Assignment	Perform Assignment
Instructional Plan	Lecture	Assessment	Lecture	Simulation
Learning Activity	Review	Study	Review	Study
Objective	Manage basic components..	Describe the networking..	Describe the networking..	Designing and implementing..
Skill	Mathematical Skills	Mathematical Skills	Logical Thinking	Critical Thinking
Cognitive Style	Field Dependence	Field Independence	Field Dependence	Field Independence
Learning Style	Diverging	Assimilating	Diverging	Diverging
Personality	Conscientiousness	Neurocritism	Conscientiousness	Conscientiousness
Interactivity	Low	Medium	Medium	Medium
Interest	Concepts	Network Operation	Network Access	Network Operation
Preference	Cognitive capabilities	Practical capabilities	Practical capabilities	Cognitive capabilities
Presentation	Text	Application	Video	Text
Difficulty Level	Easy	Medium	Medium	High
Domain Concept	5. Ethernet	1. Exploring the Network	1. Exploring the Network	2. Configuring a Network..
Knowledge Level	Intermediate	Expert	Intermediate	Intermediate
Learning Performance	Medium	Low	High	Medium
Prior Knowledge	7.1. IPv4 Network Addresses	6.1. Network Layer Protocols	2.3. Address Schemes	3.3. Moving Data in the Network
EXPECTED OUTPUT	7.2.1. ICMP	7.2.1. ICMP	4.1.2. Layer2 Frame Structure	4.2.3. LAN Topologies

Figure 5.9 Sample of learning case scenario

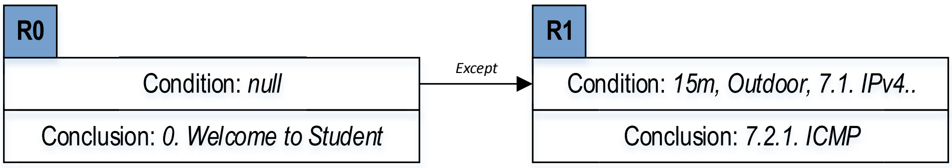


Figure 5.10 KB with two nodes

4) In the same way, Case 3 is evaluated against R0 and R1. Since the condition in R1 is false, a new rule is added as a false node for R1. The expert offers a new conclusion '*4.1.2. Layer 2 Frame Structures*' because of the condition '*45m, Library, 6.1. Network Layer Protocols*' in Case 3. These conditions and conclusion are stored as R2 (Figure 5.11). Case 3 itself is stored as the cornerstone for R2.

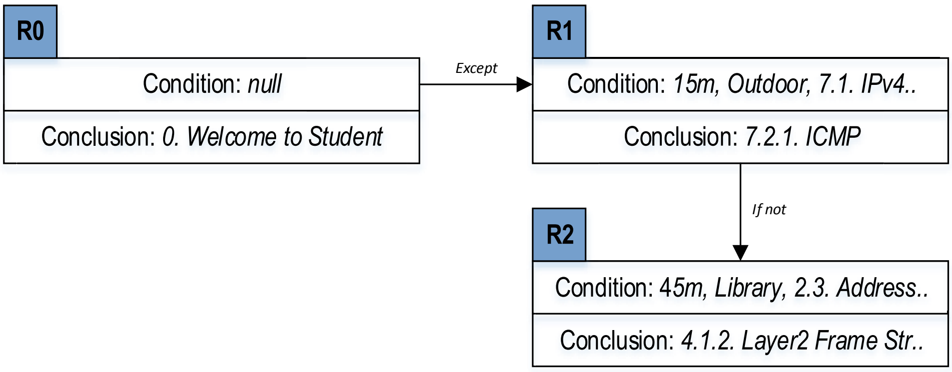


Figure 5.11 KB with three nodes

5) The last case, Case 4, is presented to the KB. After evaluating this case against the KB, the condition is found to be false against R1 and R2. Thus, a new rule is needed to handle Case 4. This rule is added as a false branch of R2. In the same process, the expert provides a new conclusion and conditions for Case 4 (Figure 5.12). The case 4 is also stored as a cornerstone case for R3.

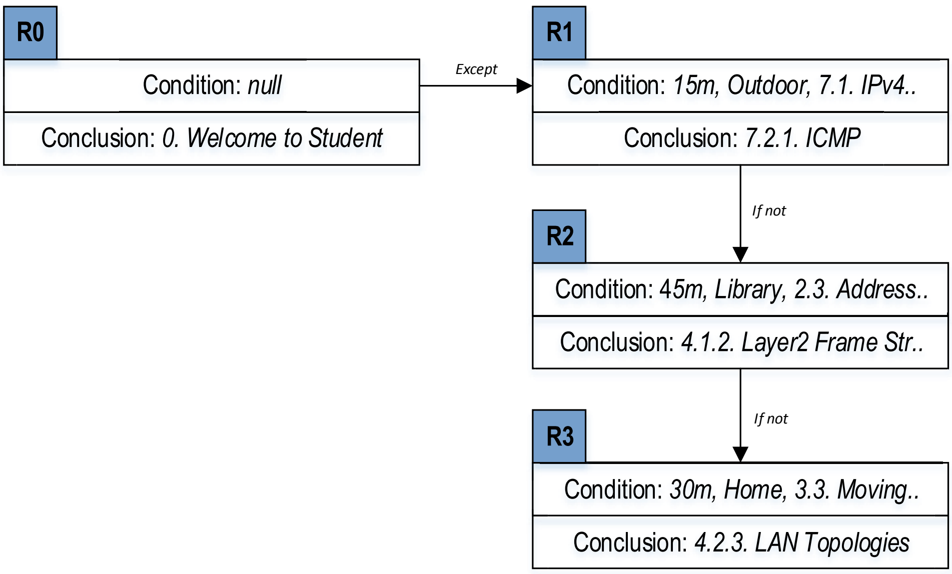


Figure 5.12 Final tree with four nodes

Although the inference mechanism seems simple, the rule refinement is indescribably complex. The case is processed sequentially with the correct classification from an expert. Minimal effort is needed to provide the criteria by an expert, as an expert only needs to compare the presented case with the rule being fired. Once the decision is made, the system adds the case to the refinement path and this process cannot be undone at a later stage. However, this knowledge is valid for the whole set of observed cases. If the KB sees a new case that was never presented before, then an adaptation is made by selecting the closest matching rule with the presented case. This approach provides an example of an adaptation mechanism which could be implemented in the e-learning domain.

5.1.3 Phase 3: Analysis of the Evaluation

Phase 3 is mainly the analysis of the developed knowledge base of the model and test cases. Statistical analysis is performed to predict the possibility of errors from the knowledge base (Beydoun & Hoffmann 2013). It also provides a measurement of KB readiness by an indication of how deeply the knowledge base was tested and how effective the test cases have been. This process facilitates the experts so that during the knowledge acquisition process, they can estimate the effectiveness of the developed knowledge base. Additionally, it helps management or executives in that it does not take long to determine the profit and loss, and hence it does not take a long time to determine whether this KB will continue to be used or not. There are at least two categories of parameters required to implement this dynamic evaluation. These are, evaluating the coverage and evaluating the predictivity. This approach can be realised by the following steps:

Step 1: Collects the base datasets

While the process of the knowledge acquisition is undertaken, the monitoring module is started to collect records from every single transaction. A record is a group of data which contains one item of information related to a cycle of KA process. In specific, the system collects the following features:

a) *Cases (n).* This variable indicates the number of test cases in the system. The value of cases (n) increases each time a case is entered into the system, whether the result of the interpretation is successful or unsuccessful.

b) *Correct cases (t).* This variable counts the cases that are correctly classified by the rules in the KB. This variable is different from the number of true branches (exceptional branch) in the KB, but it shows all the cases that are correctly classified by all rules in the KB. It means that when the expert is satisfied with the system's output, the case will be saved in the *Correct cases*.

c) *Incorrect cases (f).* In contrast with Correct cases, the *Incorrect cases* variable shows the number of cases that are incorrectly classified by the rules. When the expert inputs a case and disagrees with the system's output, the KB is said to fail. Then, the expert adds the new rules to handle the wrong conclusion. Meanwhile, the case will be collected in *Incorrect cases*. As the RDR uses failure cases to build rules in the KB, the number of Incorrect cases will be similar to the number of rules.

d) *Rules (r).* This variable shows the number of rules available in the KB. The rule that counts in this variable is a rule that is created from the result of the knowledge acquisition process, both in the true branch or the false branch. The rules (r) also reflects the size of the knowledge base.

e) *Correct Gap (gap).* This variable indicates the correct performance in a sequence of *n* instances during the KA process. The gap is marked based on the first *correct case* until an *incorrect case* is found. The process is repeated when the next *correct case* is found and so forth. Figure 5.13 illustrates the gap in a sequence of 30 cases. The first gap is found between cases 6 to 8 (3 instances), while the last gap of 5 is found between cases 22 to 30 (9 instances).

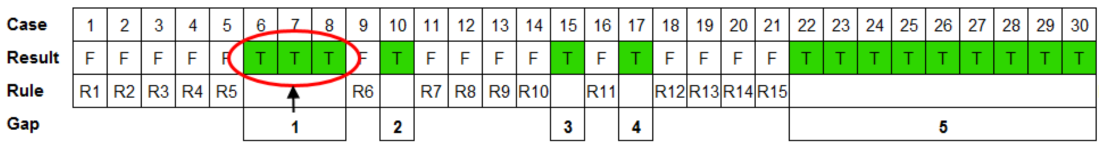


Figure 5.13 Illustration of Correct Gap

Step 2: Determine the confidence interval

Confidence interval measures the uncertainty about a characteristic of distribution with an estimation that an interval is expected to fall. A sample in random sampling does not provide perfect information about sample population. Therefore, using confidence interval one can provide a limit to claim with a specific degree of confidence. For example, a 90% confidence interval for a parameter θ means: "If one repeatedly calculates such intervals, 90% of the intervals would include the actual value θ." In this statistical analysis, the N% confidence interval is derived to estimate the coverage (r). The constant Z_N given in Table 5.3 defines the width of the smallest interval about the mean that includes two-sided N% of the total probability mass under the bell-shaped curve in a Normal distribution. In this approach, the confidence interval is one-sided and upper bound only. Thus, the user needs to decide the desired confidence level for estimating the coverage of *r*.

Table 5.3 Values of Z_N for two-sided N% confidence intervals

Confidence level N%	80%	90%	95%	98%	99%
Constant Z_N	1.28	1.64	1.96	2.33	2.58

Step 3: Processing the statistical datasets

Statistical attributes associate with each instance of time, a test or knowledge acquisition process. The value of each statistical attributes is obtained when testing the system. Statistical attributes are calculated computationally during the KA process. Based on the results of Step 1 and confidence interval in Step 2, the system calculates and presents important relationships such as coverage, potential error, standard deviation, predictivity.

a) *Coverage (Q).* This variable is used to determine the impact of the rules within the set of the knowledge base. The coverage plays an important role in measuring the effect of adding new rules on the cases of the knowledge base. The ratio of the correctly classified cases with total cases in the KB is used to estimate the coverage value. That is:

$$Q = \frac{correct\ cases\ (t)}{cases\ (n)}$$

In our case, only the upper bound of *coverage (r)* in the single sided interval of statistical theory is used. Thus, we could estimate the coverage of *r* with the ratio of *Q* and the width of smallest interval Z_M that include *M%* confidence interval of the probability mass under the bell curve normal distribution. That is:

$$Coverage(r) = Q + Z_M\sqrt{\frac{Q(1\text{-}Q)}{n}}$$

b) *Potential error of the KB (E_{KB}).* This variable reflects the level of potential error of the knowledge base. It is a measure of how likely the rules in the KB provide errors during the knowledge acquisition process. Note that the value of potential error does not reflect the true error of the knowledge base. However, it reflects the likelihood that an error will exceed the upper bound of the distribution. Thus, two components need to be considered when calculating the potential error of the KB. These are, the error due to the addition of new rule *r* in the KB (E_r) and the error for the whole KB without the new rule *r* (E_{r2}). Hence, the formula for Error of the KB (E_{KB}) is:

$$E_{KB} = E_r + E_{r2}$$

Given an upper bound δ, with n correctly classified test cases, and m_1 correctly classified cases by the last rule r. The estimations of E_r and E_{r2} respectively, are:

$$E_r = 1 - \sqrt[m1]{\delta}$$

$$E_{r2} = 1 - \sqrt[n-m1]{\delta}$$

c) *Predictivity (P).* Predictivity gives an indication of the effectiveness of the rule, that is, the fraction of cases that are evaluated by the rule in the KB. In other words, it is a measure of how valid the knowledge base in the system is, as a result of the classification process. The final goal of predictivity is to evaluate the success of the knowledge acquisition process. The lower this predictivity is, the lower is the quality of the expertise. If the predictivity is high, then the expertise is said to be more effective. The error of the KB is used to obtain the value of the Predictivity (r) as follows:

$$Predictivity(P) = 1 - E_r$$

5.2 Architecture of the Knowledge-Based System

This section describes the architecture of the knowledge-based system; to put into practice the proposed solution. This KBS is named INCAES: INCremental-based Adaptive e-learning system. INCAES is implemented as a three-tier model that provides three levels of services: *user layer, application layer,* and *data layer*. The first user layer provides interaction through the web browser and at the same time forwards the request to the application layer. The system distinguishes two user modes of operation: *learner* and *expert*. The learner mode only interacts with the learning material related to a specific course that is presented by the system. The expert mode can access all components in the application layer, including course creation and knowledge acquisition mode for creating rules. The actions of the user with the interface are passed to the application layer. The purpose of the application layer is to process the requests, commands, makes logical decisions, and perform a calculation. It also serves as a bridge that moves the data between the user layer and the storage layer. Besides, the application layer contains four main components: cases module, data model, e-learning, and monitoring. After the application layer processes the request, it sends the response back to the user layer and presents it to the user. For example, when a learner enters the learning case from a web browser, the application layer retrieves and proposes a most appropriate learning lesson from the knowledge base using adaptation rules in data layer. The overall architecture of INCAES is illustrated in Figure 5.14. It shows a connection between the user layer, application layer, and the data layer. The details of each component are discussed in the following sections.

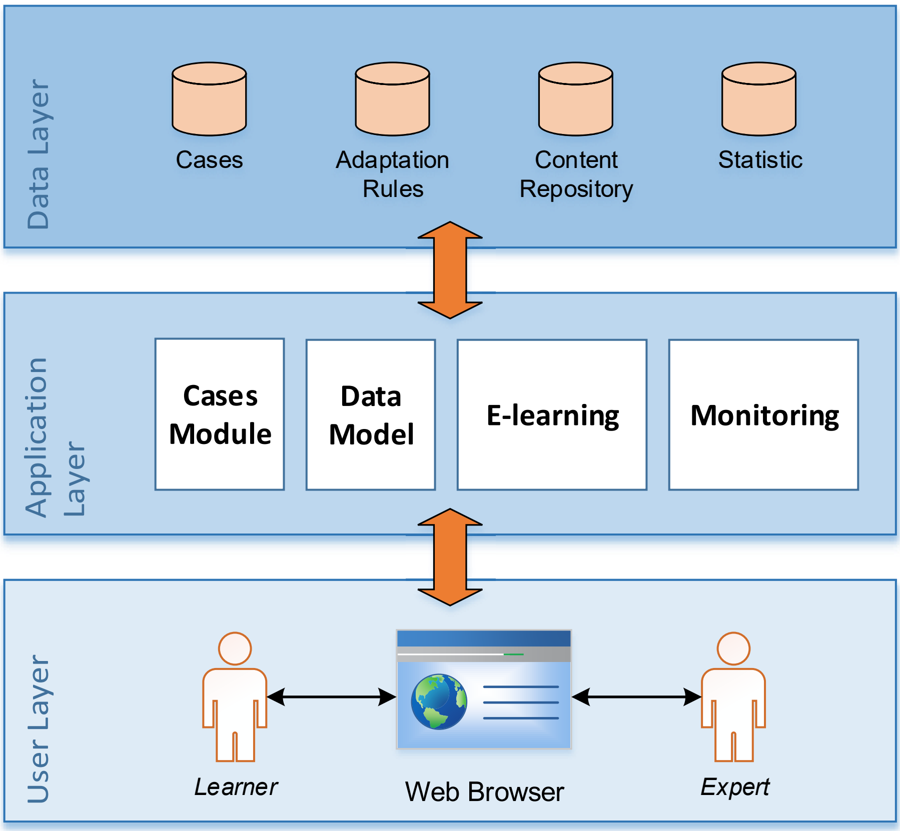


Figure 5.14 INCAES Architecture

The method for producing learning content recommendations consists of two main mechanisms: (1) *examine presented case*, which represents the student attributes during learning activity; and (2) *knowledge acquisition process* that will inspect the presented case and compare it with the rule in the knowledge base. The case is a perception of the student that interacts with the system. A case consists of a condition and a conclusion. Hence, a condition contains several attributes of the learner and their respective values. The cases module is the component that processes these two mechanisms from retrieving the presented case to the knowledge acquisition process and storing the rule in the KB. The cases module ensures compliance of the input by retrieving attributes and values of learner model from the database. This mechanism prevents the user from typing random values. Then, the data of cases is stored in several tables, including table case, rule, and feature. Figure 5.15 shows the main tables utilised by the cases module.

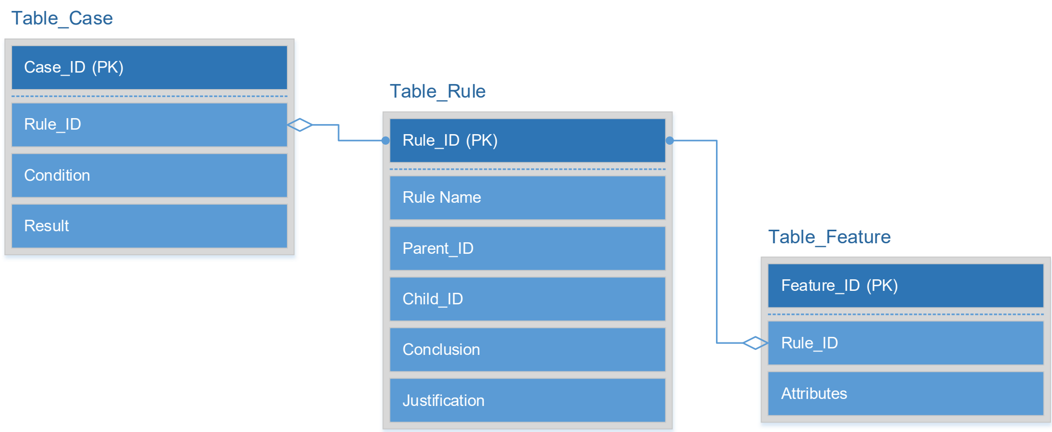
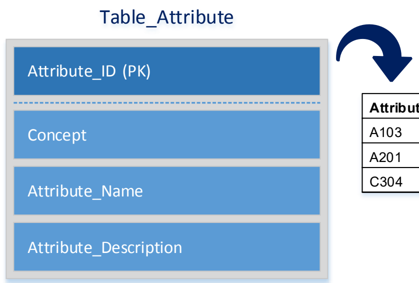


Figure 5.15 Table structure for module Cases

As shown in Figure 5.15 above, the table *Rule* has relations to table *Case* and *Feature*. The table *Case* stores information regarding condition and clause from the presented case. The conditions, which contain attributes and values, are stored in an array format in the table. If the expert disagrees with the conclusion during the KA process, then this case will be stored as a reference in making the rule. The data related to rules are stored in the table *Rule*. Furthermore, the expert should distinguish some features from the presented case that will be in the difference list for the created rule.

This feature of difference list is stored in table *Feature*.

Furthermore, the data model stores all information related to the learner model and the structure of the domain knowledge. This module has three tables, including table *Attributes*, table *Values*, and table *Output*. Table *Attributes* stores all information about attributes of the learner model. This table will be the primary reference for all processes that require the learner attribute. Each attribute of learner model has alternate values that are stored in table *Values*. The last table, *Output*, saves information related to lesson learning in e-learning. This information is used as the conclusion for the recommendation case. Figure 5.16 illustrates the sample data values for the table *Attribute*.



Attribute_ID	Concept	Attribute_Name	Attribute_Description
A103	Personal User Characteristic	Experience	Information related with the working…
A201	Personal User Characteristic	Knowledge Level	Represents a record that reflects the…
C304	System-related Characteristic	Format	Technical datatype or media type of…

Figure 5.16 Sample of data values for table Attribute

As the name implies, the purpose of the monitoring module is to monitor the knowledge base changes according to the case entered when an expert is performing a knowledge acquisition process. INCAES provides two interfaces to monitor the knowledge base. The first interface is the *KB Status*, which displays brief information related to all statuses of the knowledge base, for instance, number of cases, rules, most fired rule, and latest potential error. The second interface is *Statistical Information*, which displays the resume of measuring correct gaps from the knowledge acquisition process. This information is collected from the first case entered into the system until the latest presented case. This information is presented in a Table for further analysis by the expert.

5.3 Chapter Summary

This chapter has presented a methodology which allows integration of development and evaluation of a KBS. The advantage of this integration is to allow a KBS to be evaluated quickly during the knowledge acquisition process. Both the ripple-down rules and dynamic evaluation methodology were adapted to construct our KBS (INCAES) incrementally. At the end of the development phase, the system is tested to build a knowledge base from a series of test cases of adaptive learning. During the knowledge acquisition process, the system is evaluated directly without separating and iterating the validation activity. This process is realised by facilitating a monitoring module for inspecting the knowledge base. The domain expert can track the knowledge base development through the monitoring module that calculates a statistical analysis on the fly during KA. Then, the monitoring module displays performance of the KB for further analysis. This chapter is also concerned with the development of artefacts for validation and to complete the development of KBS (Section 5.3).

In the next chapter, the first evaluation in adaptive e-learning is conducted. It consists of a development case study in a real subject of adaptive learning, and it uses test cases from the initial learner model in Chapter 4. The development case study consisted of three phases of dynamic evaluation methodology. At the end of the case study, the initial knowledge base is validated. The experimental results are then presented and discussed for further investigation.

6 CASE STUDY 1: LEARNING CONTENT ADAPTATION FOR WEB PROGRAMMING SUBJECT

Chapter 5 has detailed the dynamic evaluation framework, which includes the design of a KBS using incremental knowledge acquisition RDR. The prototype of the KBS is developed to identify the suitable learning resources for the learner. This chapter deploys the framework in adaptive e-learning. A case study is used to evaluate the system. The learning resource for the case study is based on the Web Programming subject for the undergraduate student in IT/CS field. The KBS has stored the learning materials which is tailored with the test case entered into the system. The test case includes a pair of attributes and values that represent a learner in an adaptive learning environment. These attributes had been investigated and presented in Chapter 4. This chapter aims to implement the INCAES and investigate the performance of its KB during early development stage. The evaluation was conducted by monitoring the progress of the first ten gaps of the knowledge acquisition process. Following the evaluation of the initial knowledge base, the experiment was proceeded using the rest of the test cases to evaluate the performance of the KB.

This chapter is divided into six sections. The first section gives an overview of the experimental setup, including the purpose, procedure, method, and the scenario to evaluate the prototype. Section 6.2 presents a detailed description of the domain knowledge of AES that is domain model, learner model, and instructional model. The implementation of the framework is covered in Section 6.3. The realisation follows the guideline of the dynamic evaluation framework, as discussed in Section 5.2. Following the implementation, Section 6.4 presents the main findings and validation results obtained from the experiment. The analysis and investigation of the results are discussed in Section 6.5. Section 6.6 concludes and summarises the chapter.

6.1 Evaluation Scenario for Case Study 1

According to Design Science Research methodology, evaluating the artefact should encompass all the dimensions of the developed artefact to guarantee its efficacy and effectiveness. Therefore, this case study aims to evaluate how effective the proposed method is in taking a decision to stop the knowledge base development process and declare it sufficiently effective. A prototype of RDR KBS is built for solving adaptation issues in AES. The test case input is generated from the learner model defined in Chapter 4. In this first case study, the prototype of the KBS is designed to deliver adaptive learning contents in the subject of *Web Programming* (UTS 2018b). This subject is a core subject for the undergraduate students in Information Technology/Computer Science (IT/CS) course. The two academic experts in web programming and networking are involved not only in guiding the design of the AES but also providing the knowledge (advising) which the learning activity is recommended in a given learning environment. The collection of their advising instances become the knowledge that underpins the adaptive learning supported by the system. The experts also ensure that the artefact works as intended to address the adaptation problems i.e. they monitor the performance of the system and intervene with new knowledge if required.

The first step of this case study is to determine the models of the AES. The process begins with defining the subject and then describing the structure of *domain model* for the case study. The domain model organises the learning contents and learning activity for the students. If AES uses domain model to represent the targets or the decision for taking the adaptation, INCAES uses this information as the output of the knowledge base. The *learner model* and the *instructional model* were described following the domain model. Learner model represents a user that has several attributes which will access the educational resources through the system, while instructional model outlines the rationale behind the adaptation strategy that will be reflected through the rule set in the knowledge base. This case study follows the learning path adaptation by providing a number of learning paths that should be followed while learning, by considering several characteristics of the learner. The design of the learning path in INCAES will be configured implicitly through the selection rules of the knowledge base.

Following the description of the subject, the prototype of the KBS was designed based on incremental knowledge acquisition. The evaluation of the KBS was integrated into the development phase. Moreover, the first step in evaluation is preparing the test cases for validation. The test cases were generated by defining the parameters and values from the learner model that had been identified previously (Chapter 4). The pairwise test case model was used to generate the test cases. This method ensures that at least one possible pair of parameters will be covered in the dataset. The generated test cases were used according to their sequences. The number of test cases used in the experiment is not specified as the evaluation is based on the statistical analysis during the knowledge acquisition process. The number of the test cases used will be known after the expected estimation has been reached. Then, the performance of knowledge base is measured by analysing the growth of knowledge acquisition.

This first case study considers the decision to stop testing in the early stage of the knowledge base development process. However, there is no agreement on how early the testing should be stopped. After examining state of the art on this matter (Caraca-Valente et al. 1999), some authors proposed the number of test cases between seven to ten as early stopping for testing or evaluation. For that reason, we considered stopping testing early after reaching ten gaps in this case study. The results of the testing are presented and analysed. Further investigation is performed to determine whether the performance of the KB is acceptable for the rest of the test cases. Thus, we continue to examine the next 50 test cases without modifying any rules in the KB. It also provides greater insight into the effect of an early stop on the accuracy of the KB. The results of the investigation are presented and discussed to conclude the evaluation of the first case study.

6.2 Overview of Web Programming Subject

Web Programming is a subject for the undergraduate course in Information Technology/Computer Science (UTS 2018b). The subject introduces students to common website technologies and internet-based programming. The general concept of internet and foundations of websites are studied as well as techniques for building websites from basic HTML. The subject teaches JavaScript, JQuery, and Ajax to enhance the interfaces of web pages. It also introduces a basic understanding of how to use XML documents on the web. In the advanced level, the subject explores the use of PHP as a server-side scripting language and introduces its structure, function, and variables. PHP works best in managing websites with MySQL as the database server. Thus, the subject covers MySQL server for backend database. Finally, it teaches a sound basis of how to use framework, understands its components, and building websites using framework.

The main topics and structures covered in this subject are shown in Figure 6.1. In this structure, subject is the highest level of the course. It studies a particular topic within a course area, for example in this case study it is Web Programming. A subject is decomposed into several *modules* that contain the core ideas of the subject. Then the module is realised by several *chapters*. This defines the goals or objectives that the student must perform. The smallest element of learning is a lesson. It represents the educational content and an activity that must be performed, such as acquiring a concept, following a tutorial, taking an assignment, etc. The Web Programming subject consist of 6 modules, where each module is divided into 5-7 chapters. Overall the subject has 30 chapters and 76 lessons.

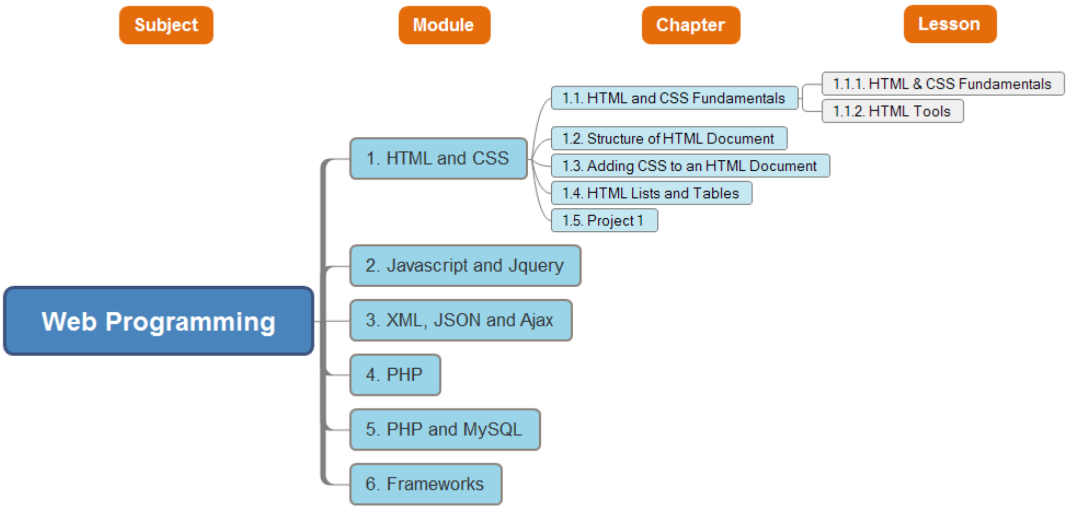


Figure 6.1 Structure of Web Programming Subject

6.2.1 Domain Model

The domain model defines the structure and the hierarchy of content materials. In AES, domain model provides the guideline for modelling the learning activities and learning resources. The structure of the domain model in this case study is represented in three hierarchical levels of abstraction: *learning goal, concepts,* and *educational materials*. Figure 6.2 illustrates the structure of the domain model in our case study. The subject distributes the learning goals through six modules. Each learning goal corresponds to a topic, which is associated with a set of concepts. Each concept is associated with educational materials consisting of lessons.

Each lesson relates to an educational content that includes a learning activity of the domain knowledge. Thus, the student learns by interacting and engaging with the representation of the educational content in the system. As the smallest element of the domain model, the educational material constitutes multiple representations of the concept, such as the reading text, multimedia content, tutorial, assessment, exercise, simulation, or assignment. In this case study, each module is organised to comprise a set of content focus, followed by a reflection. The activity in the content focus mainly is the description or explanation of the concepts through reading text. The reflection consists of a project and quiz. The project session is an activity that supports the learning by doing activities, such as a simulation, coding practice, or a design activity. Meanwhile, the quiz session contains a set of question and answer sessions. This activity helps students in reflecting their understanding about the topic.

6.2.2 Learner Model

As mentioned earlier in Chapter 4, learner model describes a user that has several attributes which will access the educational resource through the system. In AES, the learner model can be presented using several techniques. These approaches vary from *manually* presenting a value for each attribute to *automatically* identifying the learner's behaviour in the system (Brusilovsky & Peylo 2003). In this research, the learner model is manually presented and represents the stereotype of learners with their characteristics. To interface with the knowledge base, either through training or decision making, instances of the learner model are presented as feature vectors consisting of various attributes with their respective values. As these attributes represent usefulness in a learning context, during decision making, the system would recommend educational content based on the matching of learner model instance. The attributes of a learner had been investigated comprehensively in Chapter 4. The results have revealed 28 attributes, which fall under three categories: personal user characteristics, previously acquired knowledge, and system-related data user characteristics. The pair of attributes and values are used as the input for the INCAES.

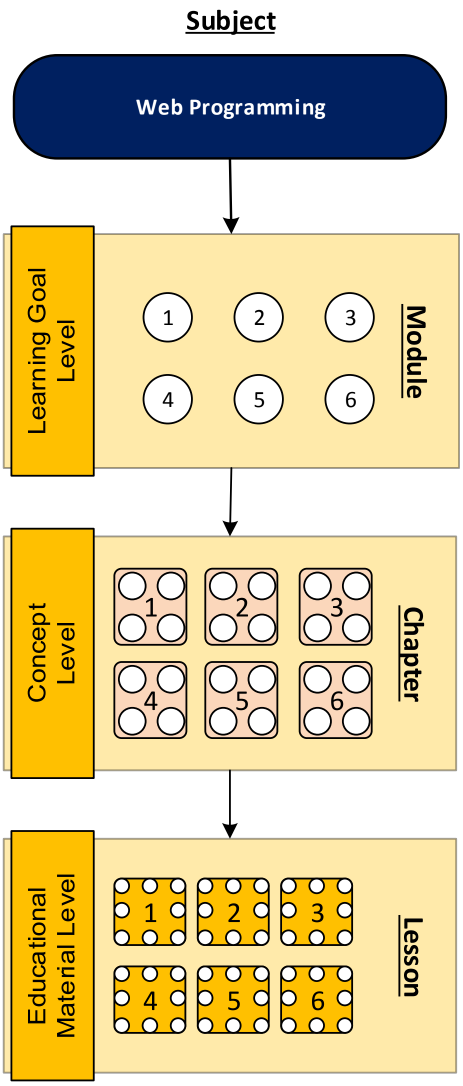


Figure 6.2 Structure of the Domain Model

The KBS processes the input and recommends educational contents for the students. During the knowledge acquisition process, the expert will provide their input on instances of the learner model. During this process, the expert observes the system's decision. If the expert disagrees with the output, they need to provide the answer and determine the attributes for the new rule and store it into the KB. This process is repeated until the disagreements between the expert and the system become far and few in between. Although the recommendations are decided by the rule set of the KB, the decision strategy is reflected by the instructional model as described in the next section.

6.2.3 Instructional Model

The instructional model contains rules and strategies to advise the relevant concepts or contents for a given learner model. As stated above, the rules are defined gradually when the test cases are input into the system. The rules are built incrementally during the knowledge acquisition process and stored in the knowledge base. If the rules contain actual notations in the system, the strategies are more like logical notions or rationale of the adaptation process. Hence, the strategies reflect the theoretical knowledge or domain expertise of adaptation in AES.

The strategies in instructional model must define how the learner model will interact with the domain model. When designing the instructional model, the experts must remember that students in adaptive learning should be able to select the learning path they prefer. AES should allow students to learn with their own plan, pace, and interest. For that reason, the strategy of adaptation is needed so that learners could progress well and finish their study. There are several strategies available for designing instructional models (Sampson, Karagiannidis & Kinshuk 2002). These are (1) offering a pre-test or post-test before selecting content; (2) analysing previous concepts or activities; and (3) suggesting topics based on the learner's interest. The expert should remember the final goal of the study to select the preferred strategy. For the first case study, the goal is set to *finish learning faster* without accessing all of the concepts. Therefore, there are three strategies in the instructional model that should be defined to achieve the goal. These are, (1) the strategy to finish the module; (2) the strategy to choose the selection of modules; and (3) the strategy to provide the possible learning paths.

In this case study, the subject of Web Programming that contains six modules, can be arranged accordingly to create adaptive content. As illustrated in Figure 6.3, the modules can be classified under three levels of knowledge: *Basic*, *Intermediate*, and *Expert*. As the name implies, this multilevel structure has prioritisation from fundamental to advance. Therefore, the modules have been ordered sequentially to help students understand the concept gradually. To illustrate the priority, the module "*6. Frameworks*" is the last module, and it only can be visited after the student has finished module "*5. PHP and MySQL*". In AES, to progress to module 5, the student must finish at least module 3 and 1. As a result, module 2 is optional for students who want to learn sequentially. If they prefer intermediate or advanced level, they can go directly to module 3. Module 5 can still be visited after finishing module 3 or 4.

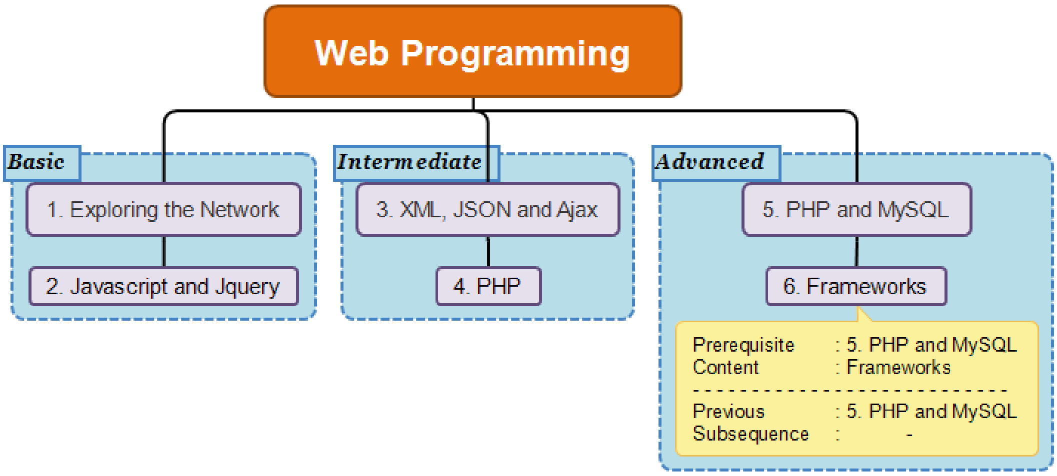


Figure 6.3 Modules Classification based on Knowledge Level

The second strategy that needs to be established is the decision to finish a module, including how and when the learners can move and finish a module. In this case study, the students can choose the lesson in any order, however they must study at least two lessons or review the summary before the module is considered completed. For instance, Figure 6.4 shows the learning flow in Module 1. The students start learning Module 1 from the first chapter "*1.1 HTML and CSS Fundamentals*". Then, they can continue in sequential order to finish the module. Alternatively, they can finish Module 1 by selecting recommended lessons in this module. The students do not have to remember all of the sequences, but the expert would structure this adaptation scenario in the system through the notions of adaptation rules.

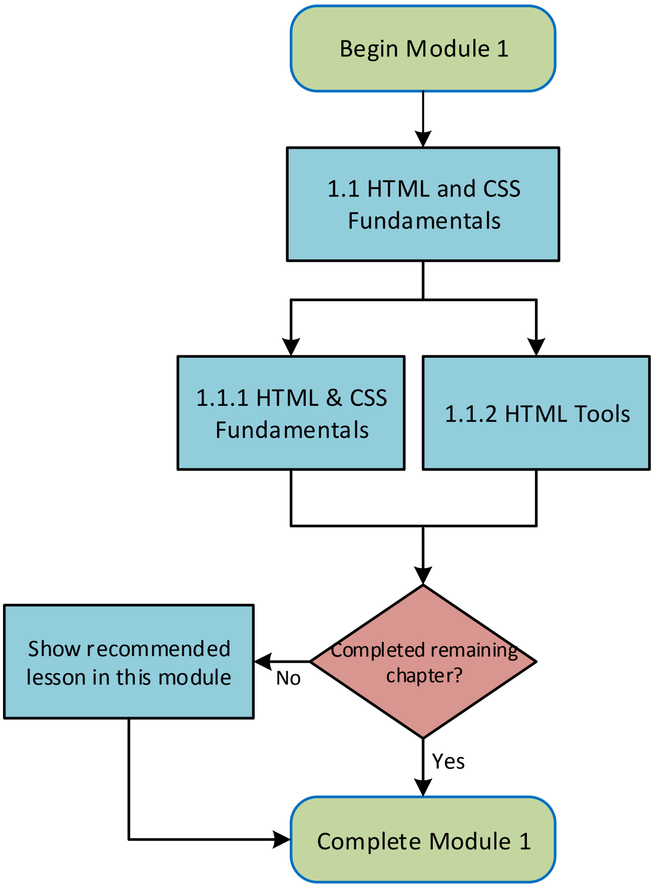


Figure 6.4 The flow to finish the Module

The last strategy in the instructional model is the strategy to provide the possible learning paths. As previously stated, the aim of adaptive learning in this case study is to finish learning faster without accessing all modules. Hence, the general strategy to advice the learning path to finish a module is "*If the student has finished a module, then check the last lesson access and their preference of the study.*" On that basis, the expert analysed the case study and suggested 21 possible output spaces for learners if they want to finish this study. These learning paths are analysed from the structure of the subject in the domain model (Figure 6.1 and 6.3), along with the learning scenarios of the instructional model (Figure 6.4).

Table 6.1 Mapping of mandatory output with the relevant attributes

No	Input Variables			Expected Output
	Domain Concept	Goal	Interest	Lesson Name
1	1. Exploring the Network	New Start	Basic	2.1.1 Javascript Fundamentals
2	1. Exploring the Network	New Start	Intermediate	3.1.1 XML Overview
3	1. Exploring the Network	New Start	Advanced	5.1.1 MySQL Fundamentals
4	1. Exploring the Network	Complete a Module	Basic | Intermediate | Advanced	1.4.1 Lists
5	1. Exploring the Network	Perform Assignment	Basic | Intermediate | Advanced	1.5.1 HTML and CSS Project
6	2. Javascript and Jquery	New Start	Basic | Intermediate	3.1.1 XML Overview
7	2. Javascript and Jquery	New Start	Advanced	5.1.1 MySQL Fundamentals
8	2. Javascript and Jquery	Complete a Module	Basic | Intermediate | Advanced	2.4.1 Events and Event Handling
9	2. Javascript and Jquery	Perform Assignment	Basic | Intermediate | Advanced	2.5.1 Tabbed Form Project
10	3. XML, JSON and Ajax	New Start	Basic | Intermediate	4.1.1 PHP Fundamentals
11	3. XML, JSON and Ajax	New Start	Advanced	5.1.1 MySQL Fundamentals
12	3. XML, JSON and Ajax	Complete a Module	Basic | Intermediate | Advanced	3.4.1 Ajax Overview
13	3. XML, JSON and Ajax	Perform Assignment	Basic | Intermediate | Advanced	3.5.1 Movie Select Project
14	4. PHP	New Start	Basic | Intermediate | Advanced	5.1.1 MySQL Fundamentals
15	4. PHP	Complete a Module	Basic | Intermediate | Advanced	4.4.1 Function Syntax
16	4. PHP	Perform Assignment	Basic | Intermediate | Advanced	4.5.1 PHP Project
17	5. PHP and MySQL	New Start	Basic | Intermediate | Advanced	6.1.1 What is a Framework
18	5. PHP and MySQL	Complete a Module	Basic | Intermediate | Advanced	5.5.1 MySQLi OOP
19	5. PHP and MySQL	Perform Assignment	Basic | Intermediate | Advanced	5.6.1 myTasks Project
20	6. Frameworks	Complete a Module	Basic | Intermediate | Advanced	6.3.4 CRUD
21	6. Frameworks	Perform Assignment	Basic | Intermediate | Advanced	6.4.1 Yii Tech Website

Furthermore, information of input variables is mapped with possible outputs of the learnt lessons. The results of the input variables (attributes and values) and expected output are illustrated in Table 6.1. It implies that the learner model with the presented values would have the content recommendation as provided. As shown in Table 6.1, some values have the vertical bar symbol (|), which is used for aliasing. It is used to facilitate the attribute which has the same value or status. As an example, in condition 4, if the student has taken domain concept *1. Exploring the Network* with the goal as *Complete a Module* and has Interest in *Basic* or *Intermediate* or *Advanced,* then the content recommendation for this condition is to continue the study with lesson *1.4.1 Lists.* This information is essential, particularly for the generation of test cases which are discussed in the following subsection.

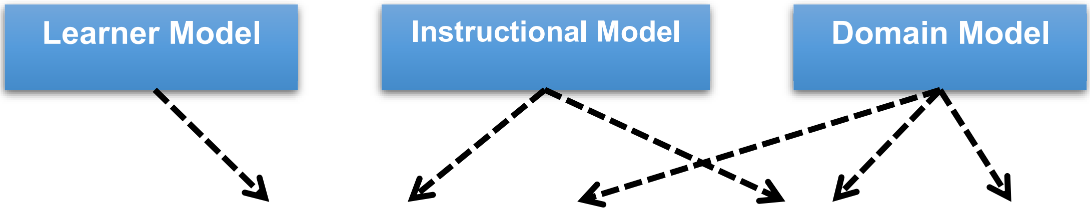
6.3 Implementing Dynamic Evaluation for Web Programming Subject

This section presents the implementation of the dynamic evaluation framework as described in Chapter 5. It encompasses three phases: (1) Construction of the Test Cases; (2) Evaluation of the Test Cases; and (3) Analysis of the Evaluation.

6.3.1 Phase 1: Construction of the Test Cases

In the first phase, the test cases are generated using the pairwise testing method. As described in Chapter 5, test cases are generated by combining the parameters and values of the learner model. In this case study, with hundred values of attributes, billions of possible test cases could be generated from this model. Thus, it is difficult to test all of them in a reasonable amount of time. Instead of attempting to cover all possible combinations, we settle on testing all possible pairs of values. Every instance of the knowledge is represented in one test case. Consequently, one test case can cover many pairs.

The first step in creating test cases is to extract values from the domain knowledge. ALS consists of three models: domain model, learner model, and instructional model. The information from each model will be used in the test cases. As an example, the information of *lesson name* from the domain model is refined to Expected output and System output. The information of the learning situation is extracted from the attribute name of learner model that is described in Chapter 4. These attributes will be the primary input variables, which consists of attribute names and values. However, to describe the learning situation, more information is needed from the instructional model. Figure 6.5 illustrates a test case that consists of Test Case ID, Input Variables, Expected Output, and System Output. While the input variables contain attributes and values that describe a learner, the expected output includes the lesson name that should be given to a student in that situation. The system output is an actual answer given by the system. This information will be recorded after execution of the test cases. Every test case reflects a learner and the knowledge that should be allocated in such a situation.



Test Case ID	Input Variables	Expected Output	System Output
1	Background: Web Technology, Competency: non-IT Certification, Course: Information Technology, Experience: Work experience, Qualification: Certificates, Bandwidth: Low, Device: PC, Emotion: Fear, Learning Motivation: Motivated, Location: Outdoor, Time: 15 m, Goal: Complete a Module, Instructional Plan: Tutorial, Learning Activity: Study, Objective: Mastering PHP coding techniques and best practices, Skill: Logical Thinking, Cognitive Style: Field Independence, Learning Style: Assimilation, Personality: Conscientiousness, Interactivity: Medium, Interest: Advanced, Preference: Practical capabilities, Presentation: Application, Difficulty Level: Difficult, Domain Concept: 6. Frameworks, Knowledge Level: Novice, Learning Performance: High, Prior Knowledge: 6.1. What is a Framework	6.3.4 CRUD	Introduction to the Course

Figure 6.5 An example of test case extraction from AES

As described in Chapter 4, we have identified 28 attributes of learner model from various sources. These 28 attributes are used as attributes in input variables. After identifying the attributes, the value of each attribute is specified by educational experts. Overall, there are 129 possible values for the first case study.

After extraction of the input variables from the learner model and the domain model, the next step is refining the case that represents the actual condition of the learning situation from an instructional model. For this purpose, an analysis is needed to define what should be included in the test parameters, because the selection of the value will determine the result of the test case. This information could be extracted from the scenario of learning in the instructional model. As explained earlier, the adaptation strategy is to finish learning early with the shortest learning path. Therefore, the test cases are built by considering learner's previous knowledge, goal and interest. Therefore, the input space is spanned by four input variables with the following value sets: *Goal, Interest, Domain Concept,* and *Prior Knowledge.*

Table 6.2 Parameters and Value Hierarchy

No	Parameters	Value Hierarchy
1	Background	Programming, Computer Networking, Human and Computer Interaction, Artificial Intelligence, Web Technology, Database, Operating System
2	Competency	IT Certification, non-IT Certification | No certification
3	Course	Information Technology
4	Experience	Work experience, Internship | Placements, No experience
5	Qualification	Certificates | Diploma, Bachelor | Graduate Certificate/Diploma, Masters | Doctoral, Others
6	Bandwidth	Low, High
7	Device	Smartphone | Tablet, PC
8	Emotion	Anger, Sadness, Happiness, Fear, Pride, Elation
9	Learning Motivation	Unmotivated, Average, Motivated
10	Location	Home | Library, Workplace | University, Outdoor
11	Time	15 m, 30 m | 45 m, 1 h | >1 h
12	Goal	New Start, Complete a Module, Complete a Lesson | Continue Last Study, Perform Assignment
13	Instructional Plan	Assessment, Lecture, Tutorial | Simulation, Case Study | Problem Statement | Project
14	Learning Activity	Study, Practise, Review
15	Objective	Building interactive websites using HTML CSS and Javascript, Designing and delivering innovative services on the Web, Mastering PHP coding techniques and best practices
16	Skill	Analytical | Critical Thinking, Logical Thinking | Mathematical Skills, Problem Solving
17	Cognitive Style	Field Dependence, Field Independence
18	Learning Style	Diverging, Assimilating, Converging, Accommodating
19	Personality	Conscientiousness, Neuroticism, Extroversion
20	Interactivity	Low, Medium, High
21	Interest	Basic, Intermediate, Advanced
22	Preference	Cognitive capabilities, Practical capabilities
23	Presentation	Video, Application, Text
24	Difficulty Level	Easy, Medium, Difficult
25	Domain Concept	1. HTML and CSS, 2. Javascript and Jquery, 3. XML JSON and Ajax, 4. PHP, 5. PHP and MySQL, 6. Frameworks
26	Knowledge Level	Novice, Intermediate, Expert
27	Learning Performance	Low, Medium, High
28	Prior Knowledge	1.1. HTML and CSS Fundamentals | 1.2. Structure of HTML Document | 1.3. Adding CSS to an HTML Document | 1.4. HTML Lists and Tables, 2.1. Javascript and JQuery Fundamentals, 2.2. Javascript Variables and Arrays | 2.3. JavaScript Syntax | 2.4. JQuery Events and Effects, 3.1 XML and JSON Fundamentals | 3.2 XML With DTD Schema | 3.3 Looping Through JSON, 3.4. Ajax Fundamentals, 4.1. PHP Fundamentals | 4.2. PHP Variables and Arrays, 4.3. Program Control | 4.4. PHP Functions and Including, 5.1. MySQL Fundamentals | 5.2. PHPMyAdmin and SQL Querying | 5.3. Connect PHP to MySQL and Fetch Data, 5.4. PHP MySQL and HTML Forms | 5.5. Object Oriented MySQLi, 6.1. What is a Framework | 6.2. Yii Framework Overview, 6.3. Gii Components

The process in test cases generation consists of two stages: *preparation* and *generation*. The preparation phase requires user to enter both pairs of the parameter (attributes) and value to be covered in the test case and also the constraints. Each combination of the parameters and values is reflected in the parameter interaction hierarchy. Table 6.2 summarises the parameters and values of the first case study. This model consists of 28 parameters and 129 values. However, the value hierarchy only has 99 entities, since some of them are aliasing to specify multiple names for a single value. For instance, the device Smartphone is treated the same as Tablet and counted as one entity. Hence, Smartphone and Tablet will be rotated among the test cases.

The next step of the preparation phase is to define the constraints. Constraints allow a user to limit the domain by specifying the unwanted combination of parameters and values. By this mechanism, the unwanted combinations can be excluded from the result, leaving the required combinations only. There are several types of constraints in this case study. The first is related to the structure of the domain model. As an example, one pair that will occur at least in one test case is the pair of Module 5 and Lesson 1.1. In reality, this combination is not possible as the learner will access Lesson 1 in Module 1, before accessing Module 5. The second is related to the learning path as previously described. The entire list of constrains are presented in Table 6.3.

Turning now to the generation process of test cases. As described in Section 5.2.1.2, an Excel-based tool named PictMaster is used to generate pairwise test cases. However, it was necessary to define some settings for this case study. The number of combined parameters used in the setting is 3. This session uses specific 4-way coverage with 90% value to ensure that the output test case will cover 90% of the domain. The generation process will be repeated three times to ensure that the output of the test cases has the desired test coverage. Finally, the process of building the test case starts with the generating of the process conforming the indicated parameters, values hierarchy, and constraint conditions. After the process is finished, the system displays statistical information to the user. Figure 6.6 shows that the tool produced 392 test cases with three repetitions, including 100% 3-way coverage and 91.1% 4-way coverage. The overall process to generate the test cases took 5 minutes and 39 seconds to get the results.

Table 6.3 List of Constraints

No	Constraints	
	Domain Concept	Goal
1	1. HTML and CSS	New Start, Complete a Module, Perform Assignment
2	2. Javascript and Jquery	New Start, Complete a Module, Perform Assignment
3	3. XML JSON and Ajax	New Start, Complete a Module, Perform Assignment
4	4. PHP	New Start, Complete a Module, Perform Assignment
5	5. PHP and MySQL	New Start, Complete a Module, Perform Assignment
6	6. Frameworks	Complete a Module, Perform Assignment
	Location	Time
7	Home	1 h
8	Workplace	30 m
9	Outdoor	15 m
	Learning Motivation	Interactivity
10	Unmotivated	Low
11	Motivated	Medium
12	Average	High
	Interest	Objective
13	Basic	Building interactive websites using HTML CSS and Javascript
14	Intermediate	Designing and delivering innovative services on the Web
15	Advanced	Mastering PHP coding techniques and best practices
	Learning Activity	Instructional Plan
16	Study	Lecture, Tutorial
17	Practise	Case Study

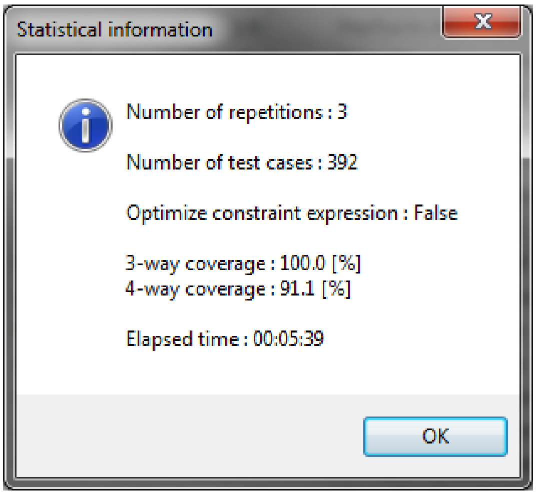


Figure 6.6 Statistical Information of First Generation

All of the test cases are generated in an excel spreadsheet. The first row contains the case attribute names, and the rest of the rows contain the cases. The total output represents a combination of input variables. In the next step, information related to the expected output from these variables is needed. This is a labor-intensive activity which requires the user to filter several attributes and add outputs in each row. According to

Table 6.1, there are four attributes that influence the generation process: Domain Concept, Prior Knowledge, Goal, and Interest. After filtering these attributes, the lesson name could be added in the expected output column. The final test cases for Case Study 1 can be seen in Figure 6.7 below. This excel-based format is imported into INCAES for further processing.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	No	Expected Output	System Output	Backgr	Compe	Course	Experie	Qualifi	Bandwi	Device	Emotio	Learnin	Locatio	Time	Goal	Instruct	Learnin	Objecti	Skill
2	1	6.3.4 CRUD		Web Tech	non-IT Ce	Informati	Work exp	Certificate	Low	PC	Fear	Motivatec	Outdoor	15 m	Complete	Tutorial	Study	Mastering	Logical Th
3	2	5.6.1 myTasks Project		Web Tech	IT Certific	Informati	No experi	Bachelor	High	Smartpho	Sadness	Average	Home	1 h	Perform A	Lecture	Review	Designing	Problem S
4	3	3.1.1 XML Overview		Programm	IT Certific	Informati	Internship	Others	Low	PC	Pride	Unmotiva	Workplace	30 m	New Start	Assessme	Review	Building i	Analytical
5	4	2.5.1 Tabbed Form Project		Artificial I	No certific	Informati	Work exp	Masters	Low	Tablet	Anger	Average	University	45 m	Perform A	Lecture	Review	Mastering	Critical Th
6	5	5.5.1 MySQLi OOP		Programm	non-IT Ce	Informati	Placemen	Doctoral	High	Smartpho	Anger	Motivatec	Library	>1 h	Complete	Case Stud	Practise	Building i	Problem S
7	6	6.3.4 CRUD		Programm	IT Certific	Informati	Work exp	Diploma	High	Tablet	Fear	Unmotiva	Workplace	30 m	Complete	Problem S	Practise	Designing	Analytical
8	7	3.1.1 XML Overview		Human an	No certific	Informati	No experi	Others	High	Smartpho	Sadness	Motivatec	University	45 m	New Start	Simulatio	Study	Designing	Problem S
9	8	2.5.1 Tabbed Form Project		Artificial I	non-IT Ce	Informati	No experi	Graduate	High	PC	Fear	Unmotiva	Outdoor	15 m	Perform A	Project	Practise	Mastering	Mathemat
10	9	6.1.1 What is a Framework		Artificial I	IT Certific	Informati	Internship	Others	Low	PC	Elation	Motivatec	Home	1 h	New Start	Tutorial	Study	Mastering	Logical Th
11	10	3.1.1 XML Overview		Programm	IT Certific	Informati	Placemen	Masters	Low	PC	Pride	Average	Outdoor	15 m	New Start	Case Stud	Practise	Designing	Mathemat
12	11	2.4.1 Events and Event Handling		Web Tech	IT Certific	Informati	Internship	Bachelor	Low	Tablet	Elation	Motivatec	Outdoor	15 m	Complete	Lecture	Study	Designing	Critical Th
13	12	1.5.1 HTML and CSS Project		Human an	IT Certific	Informati	Work exp	Certificate	High	Smartpho	Pride	Average	Outdoor	15 m	Perform A	Lecture	Study	Building i	Logical Th
14	13	4.5.1 PHP Project		Programm	No certific	Informati	No experi	Diploma	Low	PC	Pride	Motivatec	Library	>1 h	Perform A	Simulatio	Study	Building i	Problem S
15	14	4.4.1 Function Syntax		Database	IT Certific	Informati	Work exp	Others	High	PC	Fear	Average	Home	1 h	Complete	Assessme	Review	Mastering	Problem S
16	15	6.3.4 CRUD		Computer	IT Certific	Informati	No experi	Others	High	Tablet	Anger	Motivatec	Outdoor	15 m	Complete	Assessme	Review	Building i	Problem S
17	16	2.1.1 Javascript Fundamentals		Human an	non-IT Ce	Informati	No experi	Graduate	Low	Smartpho	Sadness	Unmotiva	Library	>1 h	New Start	Lecture	Review	Building i	Analytical
18	17	4.5.1 PHP Project		Artificial I	No certific	Informati	Work exp	Doctoral	Low	PC	Anger	Unmotiva	Workplace	30 m	Perform A	Problem S	Practise	Building i	Mathemat
19	18	1.4.1 Lists		Programm	IT Certific	Informati	No experi	Certificate	Low	Tablet	Elation	Unmotiva	Home	1 h	Complete	Project	Practise	Mastering	Problem S
20	19	3.5.1 Movie Select Project		Human an	non-IT Ce	Informati	Work exp	Masters	High	PC	Fear	Unmotiva	Outdoor	15 m	Perform A	Tutorial	Study	Designing	Critical Th
21	20	6.4.1 Yii Tech Website		Database	No certific	Informati	Placemen	Diploma	High	PC	Elation	Average	University	45 m	Perform A	Case Stud	Review	Designing	Analytical
22	21	5.1.1 MySQL Fundamentals		Programm	non-IT Ce	Informati	Work exp	Doctoral	High	Smartpho	Elation	Motivatec	Outdoor	15 m	New Start	Problem S	Practise	Mastering	Critical Th
23	22	3.5.1 Movie Select Project		Human an	IT Certific	Informati	Internship	Certificate	Low	PC	Sadness	Motivatec	Workplace	30 m	Perform A	Project	Practise	Mastering	Problem S
24	23	1.4.1 Lists		Programm	No certific	Informati	Placemen	Bachelor	Low	Tablet	Anger	Unmotiva	University	45 m	Complete	Simulatio	Study	Mastering	Logical Th
25	24	6.4.1 Yii Tech Website		Computer	IT Certific	Informati	Internship	Graduate	High	PC	Elation	Unmotiva	Workplace	30 m	Perform A	Lecture	Study	Mastering	Problem S
26	25	6.4.1 Yii Tech Website		Programm	non-IT Ce	Informati	No experi	Others	Low	Smartpho	Sadness	Average	Library	>1 h	Perform A	Lecture	Review	Mastering	Mathemat
27	26	2.1.1 Javascript Fundamentals		Database	No certific	Informati	No experi	Others	High	PC	Elation	Average	University	45 m	New Start	Case Stud	Practise	Building i	Problem S
28	27	5.1.1 MySQL Fundamentals		Operating	IT Certific	Informati	No experi	Masters	Low	PC	Anger	Motivatec	Home	1 h	New Start	Assessme	Review	Designing	Logical Th
29	28	6.4.1 Yii Tech Website		Database	IT Certific	Informati	Work exp	Others	High	Tablet	Happiness	Unmotiva	Library	>1 h	Perform A	Tutorial	Study	Building i	Mathemat
30	29	6.1.1 What is a Framework		Operating	non-IT Ce	Informati	Placemen	Others	High	Smartpho	Pride	Average	Outdoor	15 m	New Start	Lecture	Study	Designing	Analytical
31	30	1.4.1 Lists		Artificial I	No certific	Informati	Work exp	Doctoral	High	PC	Pride	Motivatec	Workplace	30 m	Complete	Simulatio	Review	Designing	Problem S

Figure 6.7 Example of test cases generated from PictMaster

6.3.2 Phase 2: Evaluation of the Test Case

The purpose of the second phase is developing a knowledge base from scratch by processing test cases into the INCAES. A knowledge base will be the result of the incremental acquisition of knowledge from processing the test cases. The first step of the second phase is recording the adaptive e-learning domain into the prototype as master input. The prototype received data of subject in Web Programming, which consisted of 6 modules, 30 chapters, and 76 lessons. The data was recorded in the E-learning module (Figure 6.8). Afterwards, INCAES received learner model data which consisted of 28 attributes and 129 values. The description of attributes and values is stored in the Attributes module.

The next step is processing the test cases produced by Phase 1 as input. Each case has 28 attributes with corresponding values, as well as the expected output from the case. As shown in Figure 6.8, the case can be input into the system by two ways: manually selecting the value of each attribute, or directly importing from the excel-based format. Since the generated test case is already in excel-based format, the system could recognise and enact the case automatically. After a case is loaded into the system, the inference module then detects the case and compares it with the rules in the knowledge base. The validation process involves an educational expert to check the output of the system.

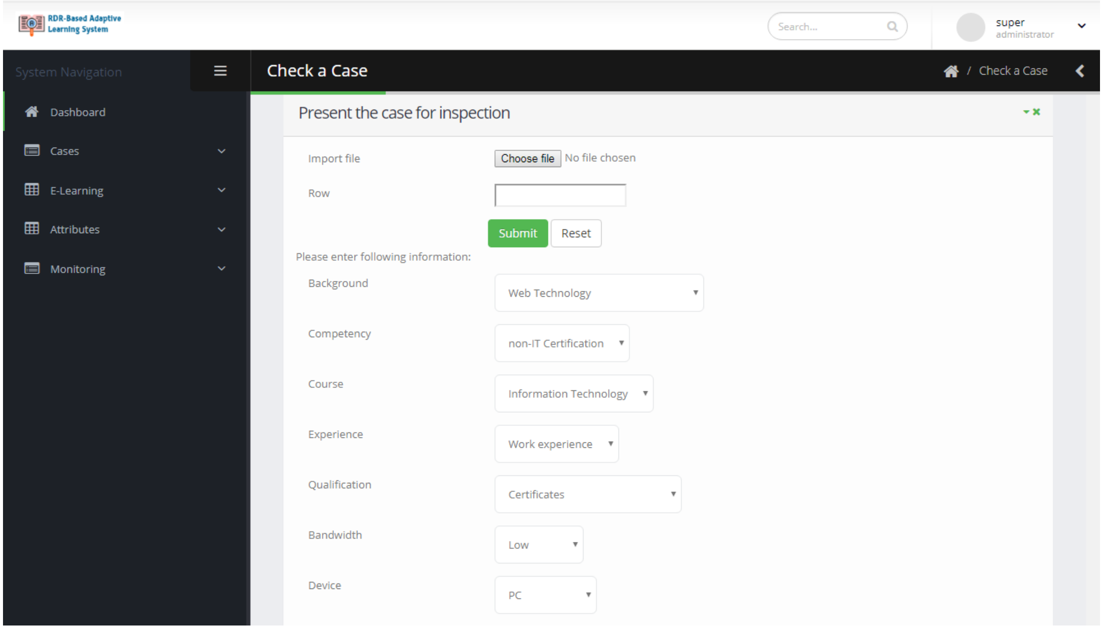


Figure 6.8 Main interface for importing attributes

We use a sample input and output for the first case as seen above in Figure 6.5. As this is the first inference, the knowledge base is still empty with only one default rule. The default rule is always true for every case. Consequently, the output of the system for the first case will be the value of the default rule, which is "*Introduction to the Course*" (Figure 6.9-A). However, the expected output for the first case should be "*6.3.4 CRUD*". Therefore, the expert disagrees with the conclusion and selects a new recommendation and gives a justification for selecting the answer if needed (Figure 6.9-B). Once the correct answer is selected, the list of attributes that differentiate the case with the cornerstone case is manually selected for the corresponding new rule (Figure 6.9-C). In this case, (1) *Goal: Complete a Module;* (2) *Interest: Advanced;* and (3) *Domain Concept: 6. Frameworks* were selected. Finally, these attributes and values (condition) and new recommendation (conclusion) were saved as a new rule of the knowledge base. The corresponding case that causes rule addition was also stored for reference as a cornerstone case.

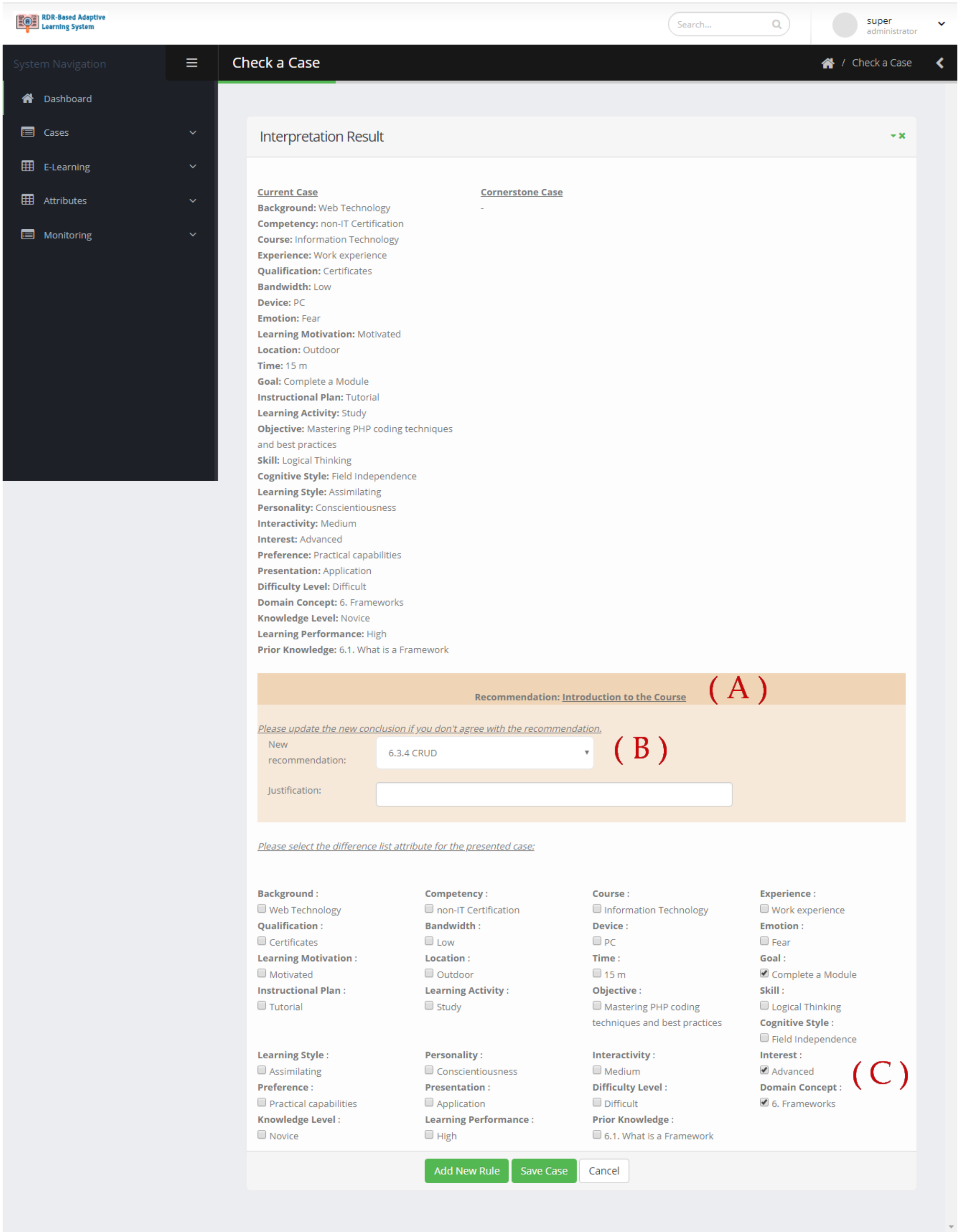


Figure 6.9 The main interface for inferencing

Once the *Add New Rule* button was clicked, the inference module created a rule consisting of a condition (learner's attributes) and conclusion (lesson name). It also stored the whole case as a cornerstone case. The process was finished in a second without further human intervention. This cycle ended the inference process and raised the knowledge base with a new rule. The next case also experienced the same treatment from the beginning to the end. Every case that entered the INCAES, will be validated directly against the knowledge base and the expert determine if the knowledge is valid. The experts only dealt with the conclusion and attributes associated with the presented case. Therefore, the cycle finished with the validated knowledge in the knowledge base. The process looks simple and sturdy due to the fact that the case was always passed through the same steps while inferencing, from representing to validation. The new rule is added only when knowledge is change, therefore modifying the system is no longer needed since the rule was ordered automatically. The above process was run iteratively until the expected performance was fulfilled. The next section discusses on how to monitor and analyse the development of the knowledge base until its ready to deliver.

6.3.3 Phase 3: Analysis of the Evaluation

For the purpose of monitoring the performance of the knowledge base, every transaction is logged and stored in the database. This log is automatically collected and transformed into a management reporting structure. This third phase is intended to identify which features of the evaluation should be focused on and how the dashboard could assist in the evaluation process. As was mentioned in the previous chapter, the first step of the third phase is collecting base datasets, including cases, correct cases, incorrect cases, rules, and correct gap. Furthermore, a statistical module computes and stores information, such as a percentage of individual rule error, domain error, knowledge base error, and predictivity. In this way the activity of Phase 3 will provide fresh insight into the evaluation phase. The current KB evaluation in AES depends on a separate phase. This is problematic since it will require more data, time and expert to evaluate the AES. This research provides an alternative to evaluating the AES by taking into account the semantic, that not heavily relies on analytical data as used in machine learning. So, the evaluation will progress along with the e-learning data growth.

Table 6.4 presents the first 10 test cases of the case study to illustrate the way data sets are collected. The table shows the expected output of the test case (2) and shows the real answer from the system (3). Every time the system gives a wrong conclusion (4), it creates a new rule to justify the case (5). A counter is used to store the number of test cases and also the correct/incorrect cases. The rule's name and position in the tree are stored in a separate table in a knowledge base. As can be seen from Table 6.4, the system started giving a correct conclusion on case 6; or after R5 was created. This occasion recurred for case 7 and 8. We recorded this correct performance in a statistical module and remarked as the gap for observation. This means that after the R5, the first gap shows the correct performance from cases 6 to 8. After repeating the experiment, we found two gaps from the first 10 cases. That is between case 6-8 and case 10.

Table 6.4 Example of the test cases

No	Expected Output	System Output	Conclusion	Rule	Parent
(1)	(2)	(3)	(4)	(5)	(6)
1	6.3.4 CRUD	Introduction to the Course	False	R1	R0
2	5.6.1 myTasks Project	6.3.4 CRUD	False	R2	R1
3	3.1.1 XML Overview	5.6.1 myTasks Project	False	R3	R2
4	2.5.1 Tabbed Form Project	5.6.1 myTasks Project	False	R4	R2
5	5.5.1 MySQLi OOP	6.3.4 CRUD	False	R5	R1
6	6.3.4 CRUD	6.3.4 CRUD	True	-	-
7	3.1.1 XML Overview	3.1.1 XML Overview	True	-	-
8	2.5.1 Tabbed Form Project	2.5.1 Tabbed Form Project	True	-	-
9	6.1.1 What is a Framework	2.5.1 Tabbed Form Project	False	R6	R4
10	3.1.1 XML Overview	3.1.1 XML Overview	True	-	-

The next steps are determining the confidence level and calculating the statistical datasets, respectively. The confidence level for this case study is set at 95% (α=0.05). The statistical datasets consist of coverage, error level, and predictivity. The coverage is the ratio of true cases by *r* to all seen cases. For the above example, the first correctly classified cases are when n = 6, until n = 8. Then the correctly classified cases are 3 or m1 = 3. Thus, the coverage value for the first gap is **Q** = 3/8 = 0.375. These values are used to calculate the potential error of the **KB**. Using a one-sided 95% confidence interval gives Z_M = 1.64. Thus, the value of E_{KB} is 0.0204 or 2.04%. The same procedure when applied to the second gap, gives us E_{KB} = 1.88%.

It should be noted that these error values are not representations of the actual error of the knowledge base. However, these values represent the correct performance based on the actual seen cases. This quick estimation was calculated as the knowledge acquisition proceeds; therefore, the next step is related to visualising this value. Overall, our monitoring system focuses mainly on the progress of the knowledge acquisition process. However, this information is consumed for two groups of users: the executive management and the domain experts. For this reason, we developed two modules, namely “KB Status” and “Expert Statistic”.

Figure 6.10 shows the KB Status module for management. This module aims to provide an overview of the knowledge acquisition project rapidly. There are two basic approaches currently being used to distinguish the features. First is an overview of the current knowledge base status. The information under this category is Total Attributes, Total Values, Total Test Cases, Total Rules, Latest Rule, and Top Rule. The second feature is the latest progress of the knowledge acquisition process. This information includes an overview of the performance from the last added rule, which are Average Cases/Rules, Error (r), Error (D), Error KB, and Predictivity.

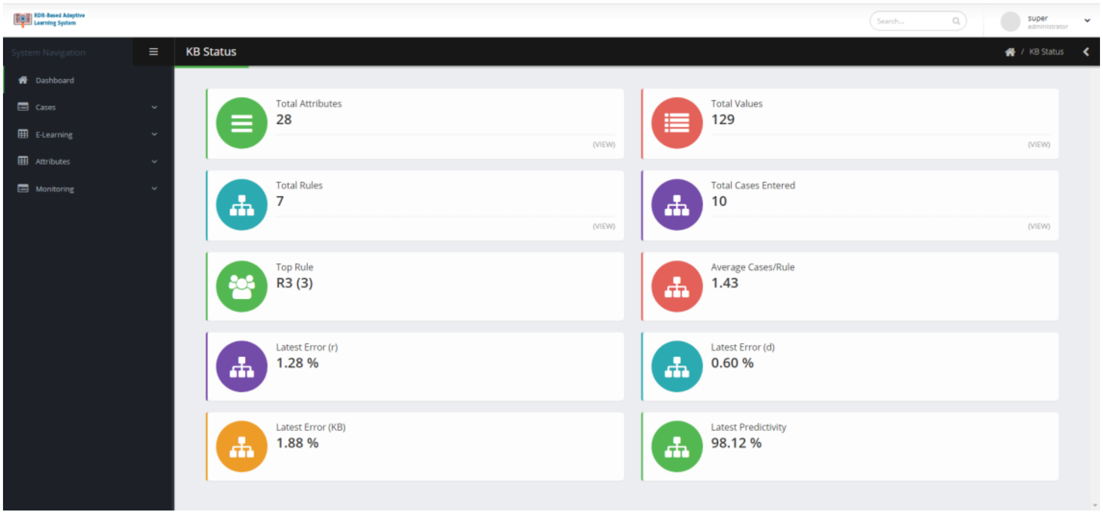


Figure 6.10 Interface of KB Status

The second monitoring module is “Expert Statistic”, which provides information to the experts on how well the RDR tree based on the test cases is seen from the acquisition process. Figure 6.11 illustrates the sequence of the latest three correct performance runs. The information in the table includes Gap, Number of Cases, Last Rule, Number of True, Standard Deviation, Error Rule (r), Error Domain (D), Error KB, and Predictivity. Each row illustrates the gap or correct performance of the knowledge base during the knowledge acquisition process. These start when the KB gives a correct answer to the query, which is then added continuously until an error is found. This provides immediate insight and flexibility to monitor the progress of the KB. Therefore, it will guide the expert to quickly and effectively determine the risk level of KB development process. The experimental results are presented in the following sections.

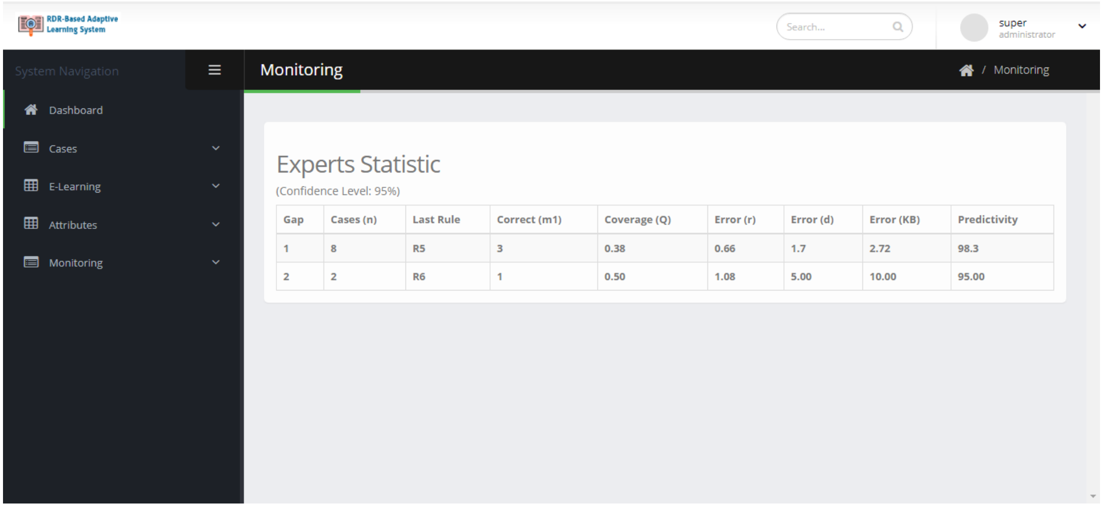


Figure 6.11 Interface of Expert Statistic

6.4 Experimental Results

Table 6.5 presents the statistical results from 10 gaps of observation. As described in Section 5.2.3, the *Ekb* expresses the estimation of maximum error that might be contained in the knowledge base. This error is broken down into two components: the errors due to a new rule *r* (*Er*) and the errors from the rest of domain without r (*Er2*). The first sequence consists of 8 cases (n) with 3 cases (m1) were correctly classified by the last added R5. Thus, the Q is 0.38 and coverage is 0.66. By using these values, the system calculates the potential error of the KB for the first gap as 2.72%. From this, 1.7% is from the last rule R5. The confidence interval for this experiment was 95% (α=0.05) one-sided upper bound. Thus, we are confident that if the test was repeated on any KB which is similar to the initial KB with the same number of test cases in gap 1, we expect that 95% of the errors due to new rule r will be less than 1.7%. It should be noted when the Er value is close to 0, it indicates a low level of error, while Er value nearly at 5% indicates a high level of error.

Table 6.5 The results of 10 gaps of correct performance

Gap	N	R	m1	Q	Cov	Er	EKB	P
1	8	R5	3	0.38	0.66	1.70%	2.72%	98.30%
2	2	R6	1	0.50	1.08	5.00%	10.00%	95.00%
3	5	R10	1	0.20	0.49	5.00%	6.27%	95.00%
4	2	R11	1	0.50	1.08	5.00%	10.00%	95.00%
5	13	R15	9	0.69	0.90	0.57%	1.84%	99.43%
6	2	R16	1	0.50	1.08	5.00%	10.00%	95.00%
7	6	R17	5	0.83	1.08	1.02%	6.02%	98.98%
8	4	R18	3	0.75	1.11	1.70%	6.70%	98.30%
9	3	R19	2	0.67	1.11	2.53%	7.53%	97.47%
10	2	R20	1	0.50	1.08	5.00%	10.00%	95.00%

After observing 10 sequences, the results indicated that the lowest *Er* was founded during gap 5, while the highest *Er* was founded in gaps 2, 3, 4, 6, and 10, respectively. We detected that these highs *Er* values were founded during the addition of the exceptional rule to the KB. Furthermore, the values of Er were in line with the predictivity P, where the highest was found in gap 5 (99.43%). It is a good indication that the KB has reached its convergence and has the capability to answer correctly. This indicator is also shown in Figure 6.12 (B) after 15 rules (gap 5) the KB has learned and started to give the correct answer. Observing the values of Er and P after gap 5, the results indicated that the values of Er and P were acceptable, although there was a high increase in gap 6 and 10. We find that these results were acceptable to demonstrate the expertise and domain for the KB. However, before taking the conclusion, it is crucial to analyse whether these results are in accordance with the actual performance of the knowledge base. Therefore, we present the actual performance results of the initial KB testing in the next section.

6.5 Performance Evaluation on Initial Knowledge Base

This section presents the performance evaluation of the initial knowledge base construction using the system. As stated before, 48 cases were applied, and 20 rules were created to find 10 gaps for analysing the first case study. Figure 6.12 shows the performance evaluation timeline for this case study. The performance evaluation was conducted by measuring the accuracy of the initial knowledge base from the first test case until the 48th case, which was the last. After the initial KB was built and initial testing result recorded, the knowledge acquisition process was halted, and no new rules were added to the knowledge base. This is to prevent knowledge modification in the KB. Then, the performance evaluation was conducted by testing the subsequent 48 cases against the initial knowledge base. Both of the results were recorded and analysed for further validation.

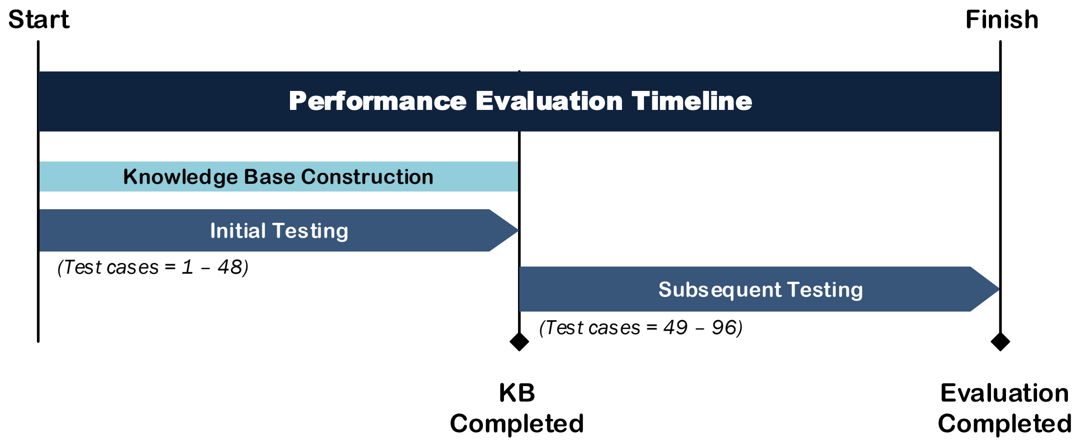


Figure 6.12 Performance evaluation plan

In order to validate the measurement of actual performance for both the initial testing and the later testing, several evaluation metrics were needed. First, we measured overall *accuracy* and used it as the basis of comparison. The overall accuracy is computed as the total number of correctly classified cases (for both classes) divided by the total number of cases. This method works well for the classification tasks with balanced class, such as for predicting the learning contents. However, the class distribution of datasets in this experiment was unbalanced or asymmetric.

For this reason, we introduced the evaluation for multi-class classification tasks by measuring *macro-averaged score of precision, recall,* and *F1*. Macro-averaged scoring methods were used since it gave equal weight to each class and worked better towards the small classes (Van Asch 2013). The method used to compute these scores was derived from the *confusion matrix* for a classification test. In particular, the Precision (P) and Recall (R) for each class C can be formulated as follows (Sokolova & Lapalme 2009):

$$P_C = \frac{\sum TP_C}{\sum TP_C + \sum FP_C}$$

and

$$R_C = \frac{\sum TP_C}{\sum TP_C + \sum FN_C}$$

where *TPc* (i.e. true positives) denotes the number of correctly classified cases as class C; *FPc* (i.e. false positives) is the number of cases that do not belong to class C but are assigned to this class incorrectly; and *FNc* (i.e. false negatives) represents the number of cases that belong to class C but which are not classified as such. F1-score (*F*) denotes a harmonic mean of precision and recall that is calculated over all entity types. It can be formulated as follows:

$$F_C = \frac{2 \times P_C \times R_C}{P_C + R_C}$$

The above formulae were used for evaluating both the initial tests and the subsequent tests.

The results presented in Table 6.6 summarise the accuracy and macro-averaged scores of precisions, recall, and F1, respectively. In general, the initial testing has average quality in terms of accuracy, with 56.25%. However, the precision, recall, and macro-F scores showed lower score quality in term of performance with each 46.18%, 49.24%, and 47.66%. In the subsequent tests, the knowledge base was constructed and hence contained enough information. Thus, the results show that the knowledge in the knowledge base was sufficient, with all measurements showing more than 80%. The measurement of both accuracy and macro-F1 shows almost similar scores: 83.33% and 83.4%, respectively. Overall, the performance scores had shown that the KB could handle better tasks when responding to the queries.

Table 6.6 Performance evaluation results of the KB (%)

Testing	Accuracy	Precision	Recall	F1
Initial	56.25	46.18	49.24	47.66
Subsequent	83.33	80.56	86.44	83.4

6.6 Chapter Summary

This chapter has presented the first case study to evaluate the proposed framework. The performance of the initial KB and final test were compared and analysed. The experiment showed that integrating knowledge acquisition and evaluation could be an alternative to evaluating the development of the KB. The results on analysing the first ten gaps of the KB showed enough data to understand the performance of the initial KB. After the initial KB was built, the second evaluation was conducted with the rest of the datasets. The performance evaluation showed that both, the initial KB and final testing F1 scores, were 47.66% and 83.4%. Although the initial KB was fair, the final testing obviously showed a good performance. The results were acceptable for the immediate evaluation since it could provide a brief information of the knowledge in the KBS. Even though the results seem promising, further evaluation is needed to understand the impact of analysing correct gaps on late stopping. For this reason, it is necessary to conduct a second evaluation with independent instructional model and domain model. In the next chapter, the experiment was conducted on a similar system. However, we proposed another subject and datasets for evaluating the system. This subject has different characteristics and instructional strategy with more variety of learning paths.

7 CASE STUDY 2: LEARNING CONTENT ADAPTATION FOR NETWORKING FUNDAMENTAL

The first case study presented in Chapter 6 was the first effort to demonstrate the RDR KBS with the supported tools. The knowledge base was instantiated with the domain knowledge from the Web Programming subject for undergraduate students. It also introduced a new way to integrate the development of KB with the evaluation. The decision to stop testing was taken early by monitoring the first ten gaps of correct performance. The tentative conclusion from the performance evaluation had shown that the KB already had adequate knowledge and was ready to use. It must be noted that this conclusion requires further investigation to verify whether the performance assessment can be generalised to any instructional model.

Furthermore, following the first case study, this chapter presents the second evaluation of the framework. In contrast to the first case study, the second case study aims to evaluate the framework with more test cases and tries to terminate the knowledge acquisition process in *the advanced development stage*. The second case study focusses on developing a knowledge base to guide dynamic learning in the Networking Essentials subject. This subject has different characteristics than before. Overall, it has 101 lessons that composed of a variety of contents and learning activities. Thus, this subject will require a different strategy to complete since it will produce a wide range of learning paths. Since the implementation of the framework has already been tested in Chapter 6, it is appropriate to revisit the steps in this chapter. A comparison of the decision to stop the knowledge acquisition between case study 1 and 2 is then discussed to conclude the framework efficacy.

The remainder of this chapter is organised as follows. Section 7.1 describes the evaluation scenario, including the goal and procedures of the evaluation. Section 7.2 briefly describes the subject of the case study, following by description of the domain model, learner model, and the instructional model. The implementation of the framework is discussed in Section 7.3. Then, section 7.4 presents and discusses the experimental results. The analysis of the evaluation and comparison of the KB is described in Section 7.5. Finally, Section 7.6 concludes and discusses the chapter.

7.1 Evaluation Scenario for Case Study 2

The previous case study has introduced the dynamic evaluation approach to build and evaluate the KB by validating it with a case study in the Web Programming subject. This chapter extends the case study to observe the impact when we take a decision to stop the knowledge base construction in the advanced development stage. For this, the case study of the Networking Fundamental (UTS 2018a) is used. This subject will be the basis of our second scenario since it offers more contents and learning activities. With more extensive resources, it will need a different adaptation strategy to complete the subject. It also produces a more possible learning output for the student. These conditions are required for the evaluation since the second case study should contain more possible test cases. The same process of implementation from generating test cases, evaluation of the test cases, and monitoring the error of knowledge base will be carried out in this case study.

The second case study proposes to examine the effect of stopping the evaluation in the advanced stage of knowledge base development. This evaluation aims to analyse whether late stopping has less impact compared to early stopping. Therefore, the second case study observes the long gaps of the knowledge acquisition process and then compares the results with the first case study. For this purpose, we increased the gaps for analysis from 10 to 25 gaps in this second case study. As previously presented in Chapter 6, the second evaluation was conducted to determine the performance of the knowledge base by examining the subsequence test cases. The final analysis was carried out to compare the results between case study 1 and 2.

7.2 Overview of the Network Fundamentals Subject

The Network Fundamentals subject for undergraduate students in IT/CS course is selected for the second case study in this thesis since it provides more learning contents and activities. This subject is preferred because with more contents, it will allow deeper insight into more possible learning path scenarios for students. The Network Fundamentals subject (UTS 2018a) is generally delivered for the second/third year of the undergraduate students. The primary focus of the subject is introducing the structure of computer networks and guiding for the data transfer process. The subject explores various components and services of the network based on OSI layer protocol. It begins by introducing networks and internet as well as the application layer that works on the internet. The main principles of data transport are studied with focuses on TCP and UDP. The subject covers the core of the network layer, from switching, internet protocol, addressing, sub netting, and routing algorithms. The techniques of switching, services, and implementation are explored further in link layer.

Figure 7.1 illustrates the main topics covered in this subject. As shown in the figure, this subject has more learning contents and variations of activity compared to the previous case study. Overall, there are 10 modules, 39 chapters, and 101 lessons offered in this subject. Consequently, this subject will take longer to finish with slightly different variety on learning paths. In this subject, each module is comprised of a set of theory content, follow by a tutorial and assignment. The theory is delivered in various forms, such as the text description, an interactive video, or structured PDF documents. The tutorial session includes learning by doing activities, simulations, or network design activity. Meanwhile, the assignment session consists of question and answer sessions with immediate feedback. It is expected that students will gain theoretical and practical experience after completing the subject.

7.2.1 Domain Model

As previously described in Chapter 6, the domain model defines the content material and the way it is structured. For generalisability, the structure of the domain model is similar to the previous case study which consist of three hierarchy levels of abstraction: learning goal level, concept level, and educational material level. The learning goals have several modules in it and act as the highest level of the domain model. The modules have chapters which are associated with a number of concepts. Then, the concepts are distributed to student through education materials, which is the lowest level of the abstraction hierarchy, and also the smallest element in the domain conceptualisation.

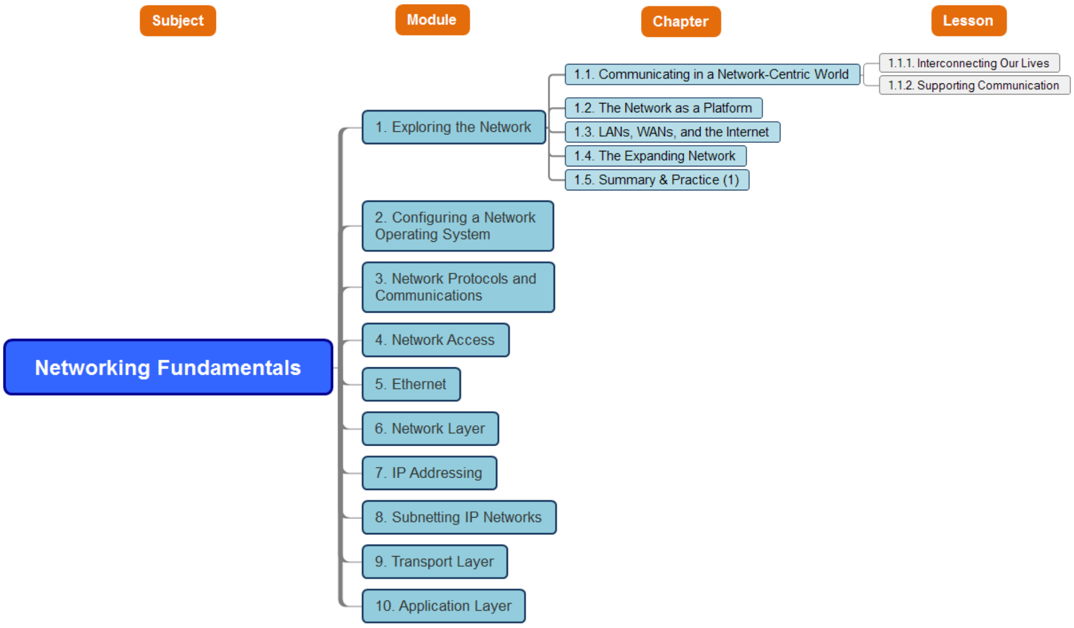


Figure 7.1 Structure of Networking Fundamentals Subject

Although the hierarchy of the domain model is similar to the previous case study, the structure of the learning activity is slightly different. A module in this case study is arranged to include activities that involve student interaction with a focus on text reading (content) or engaging with multimedia (interaction). Each module has a summary that highlights the important concepts and concludes the lecture. Even though it omits details, a summary helps students to focus and understand the entire lesson. It forces the students to follow the objectives of the module rather than following each sentence on reading text. The last activity in the module is a practice activity which consists of an assessment for evaluating the knowledge of the student. The practice presents the student with a problem, case, scenario, or design issue. To deal with the problem, students are needed to have basic knowledge and understanding of the topics.

7.2.2 Learner Model

The process of designing the learner model is similar to the one presented in Case

Study 1. The learner model is presented manually as the test case with attributes and associated values. Each of the test cases portrays a stereotype of a student with their characteristics. The attributes for Case Study 2 are also similar to those in Case Study 1. However, the values are slightly different for the attributes that relate to domain knowledge, due to the different type of learning content. The total number of values in this case study is 142, which is higher than the previous case study. As a result, the combination of inputs will be higher than the combination produced before. This is needed for evaluation purposes since the testing scenario for the second case study will be longer than the previous evaluation.

7.2.3 Instructional Model

With the same goal, the instructional model in this case study is arranged so that students can finish learning without visiting all lessons. However, the strategy needed for adaptation is different since the classification of the modules is based on the theme of topics. Some modules have been arranged equally according to their respective theme, regardless of their level of difficulty. As illustrated in Figure 7.2, all of the 10 modules are organised according to five topics as follows: *Concepts, Software, Hardware, Communication*, and *Connectivity*. An example for adaptation, the sequence of module "*9. Transport Layer*" is after module 8 and before module 10. However, according to the topics this module has no prerequisite and it could be learned immediately between any modules. This condition is different in the case of module 8, which is the last part of *Communication* cluster. To learn this module, the student must pass module 7, which also means that the student must finish learning module 6 as the prerequisite of the module 7.

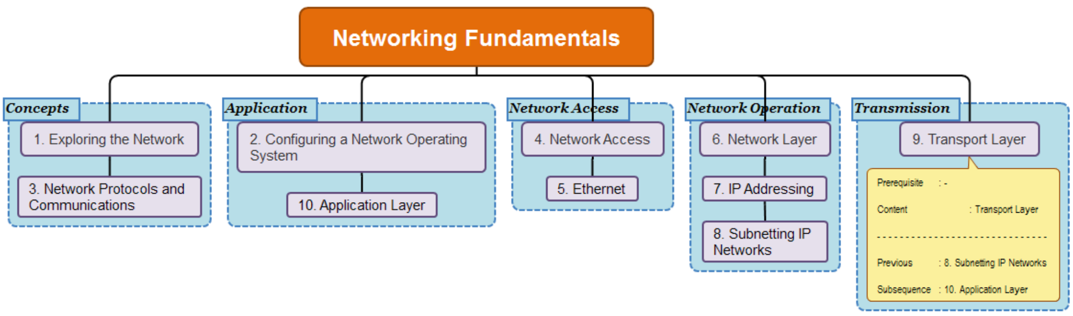


Figure 7.2 Modules Classification based on Topics

Furthermore, the strategy to finish a module in this case study is that the students can select any lesson in any order. However, they must study at least two lessons in a module and then study the summary before the module is considered completed. To design the possible learning path in this case study, we looked for the path related with the adaptation scenario where the learners get recommendations of learning content based on their goals, interests, and previous lesson. The general rule to select the best learning path after the student has finished a module is: *If the student has finished a module, then check the last lesson access and the goal of the study.* Hence, the input space is spanned by four input variables with the following value sets: Domain Concept, Goal, and Interest. With assistance from an educational expert, we have analysed the case study and found the possible 34 output spaces for learners if they want to finish this study. The details of the expected output space in this case study are presented in the following Table 7.1.

Table 7.1 Mapping of mandatory output (lesson) with the relevant attributes (values)

No	Input Variables			Expected Output
	Domain Concept	Goal	Interest	Lesson Name
1	1. Exploring the Network	New Start	Application	2.1.1. Cisco IOS
2	1. Exploring the Network	New Start	Concept	3.1.1. Protocols
3	1. Exploring the Network	Complete a Module	Concepts | Application | Network Access | Network Operation | Transmission	1.3.1. Components of Network
4	1. Exploring the Network	Perform Assignment	Concepts | Application | Network Access | Network Operation | Transmission	1.5.2. Quiz 1
5	2. Configuring a Network Operating System	New Start	Application	10.1.1. Application, Session, and Presentation
6	2. Configuring a Network Operating System	New Start	Network Access	4.1.1. Layer 2 Introduction
7	2. Configuring a Network Operating System	Complete a Module	Concepts | Application | Network Access | Network Operation | Transmission	2.3.1. Ports and Addresses
8	2. Configuring a Network Operating System	Perform Assignment	Concepts | Application | Network Access | Network Operation | Transmission	2.4.2. Class Activity - Tutor Me

9	2. Configuring a Network Operating System	New Start	Concepts | Network Operation | Transmission	3.1.1. Protocols
10	3. Network Protocols and Communications	Complete a Module	Concepts | Application | Network Access | Network Operation | Transmission	3.3.1. Data Encapsulation
11	3. Network Protocols and Communications	Perform Assignment	Concepts | Application | Network Access | Network Operation | Transmission	3.4.2. Quiz - 3
12	3. Network Protocols and Communications	New Start	Concepts | Application | Network Access | Network Operation | Transmission	4.1.1. Layer 2 Introduction
13	4. Network Access	Complete a Module	Concepts | Application | Network Access | Network Operation | Transmission	4.4.1. Copper Cabling
14	4. Network Access	Perform Assignment	Concepts | Application | Network Access | Network Operation | Transmission	4.5.2. Class Activity - Linked In!
15	4. Network Access	New Start	Concepts | Application | Network Access | Transmission	5.1.1. Ethernet Operation
16	4. Network Access	New Start	Network Operation	6.1.1. Network Layer in Communication
17	5. Ethernet	Complete a Module	Concepts | Application | Network Access | Network Operation | Transmission	5.3.1. Switching
18	5. Ethernet	Perform Assignment	Concepts | Application | Network Access | Network Operation | Transmission	5.4.2. Class Activity - MAC and Choose
19	5. Ethernet	New Start	Concepts | Application | Network Access | Network Operation | Transmission	6.1.1. Network Layer in Communication
20	6. Network Layer	Complete a Module	Concepts | Application | Network Access | Network Operation | Transmission	6.3.2 Router Bootup
21	6. Network Layer	Perform Assignment	Concepts | Application | Network Access | Network Operation | Transmission	6.5.2. Lab - Building a Switch and Router Network
22	6. Network Layer	New Start	Transmission	9.1.1. Transportation of Data
23	6. Network Layer	New Start	Concepts | Application | Network Access | Network Operation	7.1.1. IPv4 Address Structure
24	7. IP Addressing	Complete a Module	Concepts | Application | Network Access | Network Operation | Transmission	7.2.2. Testing and Verification

25	7. IP Addressing	Perform Assignment	Concepts | Application | Network Access | Network Operation | Transmission	7.3.2. Lab - Mapping the Internet
26	7. IP Addressing	New Start	Concepts | Application | Network Access | Network Operation | Transmission	8.1.1. Network Segmentation
27	8. Subnetting IP Networks	Complete a Module	Concepts | Application | Network Access | Network Operation | Transmission	8.2.1. Structured Design
28	8. Subnetting IP Networks	Perform Assignment	Concepts | Application | Network Access | Network Operation | Transmission	8.3.2. Lab – Calculating IPv4 Subnets
29	8. Subnetting IP Networks	New Start	Concepts | Application | Network Access | Network Operation | Transmission	9.1.1. Transportation of Data
30	9. Transport Layer	Complete a Module | Complete a Lesson | Continue Last Study	Concepts | Application | Network Access | Network Operation | Transmission	9.2.1. TCP Communication
31	9. Transport Layer	Perform Assignment	Concepts | Application | Network Access | Network Operation | Transmission	9.3.2. Simulation - TCP and UDP Communication
32	9. Transport Layer	New Start	Concepts | Application | Network Access | Network Operation | Transmission	10.1.1. Application, Session, and Presentation
33	10. Application Layer	Complete a Module | Complete a Lesson | Continue Last Study	Concepts | Application | Network Access | Network Operation | Transmission	10.2.1. Common Application Layer Protocols
34	10. Application Layer	Perform Assignment	Concepts | Application | Network Access | Network Operation | Transmission	10.3.2. Simulation - Implement Services

7.3 Dynamic Evaluation of the Instructional Knowledge for Networking Fundamentals Subject

The process of dynamic evaluation in the second case study remains unchanged. However, the evaluation used a different domain knowledge. The subject of Networking Fundamentals is used to evaluate the second case study.

7.3.1 Phase 1: Construction of the Test Cases

Following the proposed framework, this phase focuses on the test case creation process. The test cases creation starts by extracting domain knowledge, which in this case study is a learner model from Chapter 4 and domain model data from Networking Fundamentals subject. As mention earlier, this subject has more educational content than previous case study, where the total available values for this case are 142 and 28 attributes.

The first step towards formalising the pairwise testing is to define the parameters along with the values and the constraints. In this case study, all 28 attributes of the data model will become the test parameters. After this step, the identified parameters are decomposed into a collection of parameters values. The values of the parameter in testing is equivalent to the value of the attributes in the data model. However, an analysis is needed to define what should be included in the test parameters because the selection of the value will determine the result of test.

The first step requests user to enter the parameters and value hierarchy. Table 7.2 summarises the parameters and value hierarchy of the case study. This model consists of 28 parameters and 142 value hierarchies. However, the value hierarchy only has 102 entities as some of them are aliasing. Aliasing is a way of specifying multiple names for a single value. For instance, the device Smartphone is treated the same as Tablet and counted as one entity. So, Smartphone and Tablet will be rotated among the test cases.

Table 7.2 Parameters and Value Hierarchy

No	Parameters	Value Hierarchy
1	Background	Programming, Computer Networking, Human and Computer Interaction, Artificial Intelligence, Web Technology, Database, Operating System
2	Competency	IT Certification, non-IT Certification | No certification
3	Course	Information Technology
4	Experience	Work experience, Internship | Placements, No experience
5	Qualification	Certificates | Diploma, Bachelor | Graduate Certificate/Diploma, Masters | Doctoral, Others
6	Bandwidth	Low, High
7	Device	Smartphone | Tablet, PC
8	Emotion	Anger, Sadness, Happiness, Fear, Pride, Elation
9	Learning Motivation	Unmotivated, Average, Motivated
10	Location	Home | Library, Workplace | University, Outdoor
11	Time	15 m, 30 m | 45 m, 1 h | >1 h
12	Goal	New Start, Complete a Module, Complete a Lesson | Continue Last Study, Perform Assignment

13	Instructional Plan	Assessment, Lecture, Tutorial | Simulation, Case Study | Problem Statement | Project
14	Learning Activity	Study, Practice, Review
15	Objective	Understanding the characteristics and functions of each layer of the OSI model, Describe the networking processes within and between networking hardware, Manage basic components of a Cisco router and a Cisco switch, Apply VLSM Addressing to an Internet Protocol v4, Designing and implementing a hierarchical IPv4 addressing scheme, Recognise and make suitable choices of physical networking equipment for a small network
16	Skill	Analytical | Critical Thinking, Logical Thinking | Mathematical Skills, Problem Solving
17	Cognitive Style	Field Dependence, Field Independence
18	Learning Style	Diverging, Assimilating, Converging, Accommodating
19	Personality	Conscientiousness, Neuroticism, Extroversion
20	Interactivity	Low, Medium, High
21	Interest	Basic, Intermediate, Advanced
22	Preference	Cognitive capabilities, Practical capabilities
23	Presentation	Video, Application, Text
24	Difficulty Level	Easy, Medium, Difficult
25	Domain Concept	1. Exploring the Network, 2. Configuring a Network Operating System, 3. Network Protocols and Communications, 4. Network Access, 5. Ethernet, 6. Network Layer, 7. IP Addressing, 8. Subnetting IP Networks, 9. Transport Layer, 10. Application Layer
26	Knowledge Level	Novice, Intermediate, Expert
27	Learning Performance	Low, Medium, High
28	Prior Knowledge	1.1. Communicating in a Network-Centric World | 1.2. The Network as a Platform | 1.3. LANs WANs and the Internet | 1.4. The Expanding Network, 2.1. IOS Bootcamp | 2.2. Getting Basic | 2.3. Address Schemes, 3.1. Network Protocols and Standards | 3.2. Using Requests for Comments | 3.3. Moving Data in the Network, 4.1. Data Link Layer | 4.2. Media Access Control | 4.3. Physical Layer | 4.4. Network Media, 5.1. Ethernet Protocol | 5.2. Address Resolution Protocol | 5.3. LAN Switches, 6.1. Network Layer Protocols | 6.2. Routing | 6.3. Routers | 6.4. Configuring a Cisco Router, 7.1. IPv4 Network Addresses | 7.2. Connectivity Verification, 8.1. Subnetting an IPv4 Network | 8.2. Addressing Schemes, 9.1. Transport Layer Protocols | 9.2. TCP and UDP, 10.1. Application Layer Protocols | 10.2. Well-Known Application Layer Protocols and Services

Following the parameter and value descriptions, the next step is to input the constraints to limit the unwanted combination of outputs. The violating test cases cannot simply be removed from the result as they might cover other possible valid pairs. Instead of losing valid pairs, the pairwise testing eliminates disallowed combinations through a constraints mechanism. Thus, the unwanted combinations were excluded from the results, which left the required combinations only. PictMaster uses *If* and *Then* relations to express constraint conditions and targets. *If* constraint condition is specified, *Then* constraint target will be generated. The constraint expression applied for the model is shown in Table 7.3.

Table 7.3 List of Constraints

No	Constraints	
	Domain Concept	Goal
1	1. Exploring the Network	Complete a Module, Perform Assignment
2	2. Configuring a Network Operating System	New Start, Complete a Module, Perform Assignment
3	3. Network Protocols and Communications	New Start, Complete a Module, Perform Assignment
4	4. Network Access	New Start, Complete a Module, Perform Assignment
5	5. Ethernet	New Start, Complete a Module, Perform Assignment
6	6. Network Layer	New Start, Complete a Module, Perform Assignment
7	7. IP Addressing	New Start, Complete a Module, Perform Assignment
8	8. Subnetting IP Networks	New Start, Complete a Module, Perform Assignment
9	9. Transport Layer	New Start, Complete a Module, Complete a Lesson, Perform Assignment
10	10. Application Layer	Complete a Module, Complete a Lesson, Perform Assignment
	Location	Time
11	Home	1 h
12	Workplace	30 m
13	Outdoor	15 m
	Learning Motivation	Interactivity
14	Unmotivated	Low
15	Motivated	Medium
16	Average	High
	Interest	Objective
17	Concept	Understanding the characteristics and functions of each layer of the OSI model
18	Application	Recognise and make suitable choices of physical networking equipment for a small network
19	Network Access	Manage basic components of a Cisco router and a Cisco switch
20	Network Operation	Apply VLSM Addressing to an Internet Protocol v4, Designing and implementing a hierarchical IPv4 addressing scheme
21	Transmission	Describe the networking processes within and between networking hardware
	Learning Activity	Instructional Plan
22	Study	Lecture, Tutorial
23	Practice	Case Study

The last step of generation required user to configure the settings. Settings menu, as shown in Figure 7.3, contains all the information related to test configurations. In this model, we select "*Use constraints table*" to apply constraints to the test case. "*Optimize constraint expression*" allows PictMaster to optimise the constraint expressions that are generated from the constraints table. It means that, if the system determined that it will take a long time to generate test cases due to inappropriate constraints, the format of constraints will be optimised automatically to complete the generation in less time. "*Show statistical information*" will display information related to the frequency of generation, number of created test cases, and time to complete generation once the process has finished. Meanwhile, "*Show coverage*" displays combination coverage proportions which have been created during the test cases generation. In this case study, the specific 4-way coverage with 90% value was used to ensure that output test case would cover 90% of the domain. The generation process will be repeated five times to ensure that the output of the test cases have the desired test coverage.

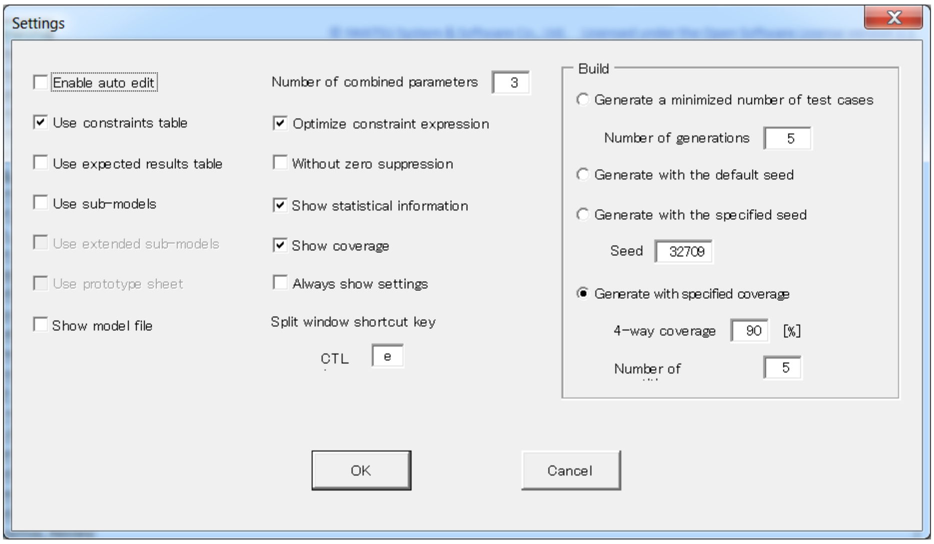


Figure 7.3 Settings Menu

The final step requires the user to build the test case, and then the tool will automatically start the generating process conforming the indicated parameters, values hierarchy, and constraint conditions. After the process is finished, the system displays statistical information to the user. The tool produced a total of 753 test cases with five repetitions. This included 100% 3-way coverage and 89.9% 4-way coverage. The overall process took 13 minutes and 20 seconds to gather the results. In the next phase, the test cases were evaluated to develop a knowledge base.

7.3.2 Phase 2: Evaluation of the Test Cases

The development of knowledge base starts from a default rule that always gives a conclusion to any condition. The test cases data is entered into the system from the first case and incrementally added to the knowledge base. As RDR merges the knowledge acquisition and maintenance process, the rules are then built based on the inference of the test cases. So, the representation of the test case is important in the RDR. Table 7.4 illustrates the first test case of the case study. The input consisted of 28 attributes and values, with an expected output for this condition. The expected output has been described previously by educational experts based on certain features on the condition of the case i.e. Goal, Interest, and Domain Concept.

Table 7.4 First test case of case study

Case No.	Input	Expected Output
1	Background: Programming, Competency: non-IT Certification, Course: Information Technology, Experience: Internship, Qualification: Bachelor, Bandwidth: High, Device: PC, Emotion: Fear, Learning Motivation: Motivated, Location: Workplace, Time: 30 m, Goal: Complete a Module, Instructional Plan: Assessment, Learning Activity: Review, Objective: Apply VLSM Addressing to an Internet Protocol v4, Skill: Analytical, Cognitive Style: Field Dependence, Learning Style: Diverging, Personality: Extroversion, Interactivity: Medium, Interest: Network Operation, Preference: Cognitive capabilities, Presentation: Text, Difficulty Level: Easy, Domain Concept: 6. Network Layer, Knowledge Level: Expert, Learning Performance: Low, Prior Knowledge: 6.1. Network Layer Protocols	6.3.2 Router Bootup

As predicted, the inference result for the first test case would be the conclusion of the default rule. As shown in Figure 7.4, the system recommends 'Message to the Student' for the first test case. To save this case as a rule, several attributes from the presented case should be selected as the feature that distinguishes this case from the others. For this purpose, we only need to analyse three features i.e. goal, interest, and domain knowledge. The domain knowledge of this case is '6. Network Layer' and the goal is 'Complete a Module'. As indicated previously in Table 7.1 (20), if these conditions with any value of interest are found in a case, then the expert would recommend '6.3.2 Router Bootup' for students. Thus, these two features (6. Network Layer and Complete a Module) and a conclusion (6.3.2 Router Bootup) must be selected to process the first test case and create a new rule.

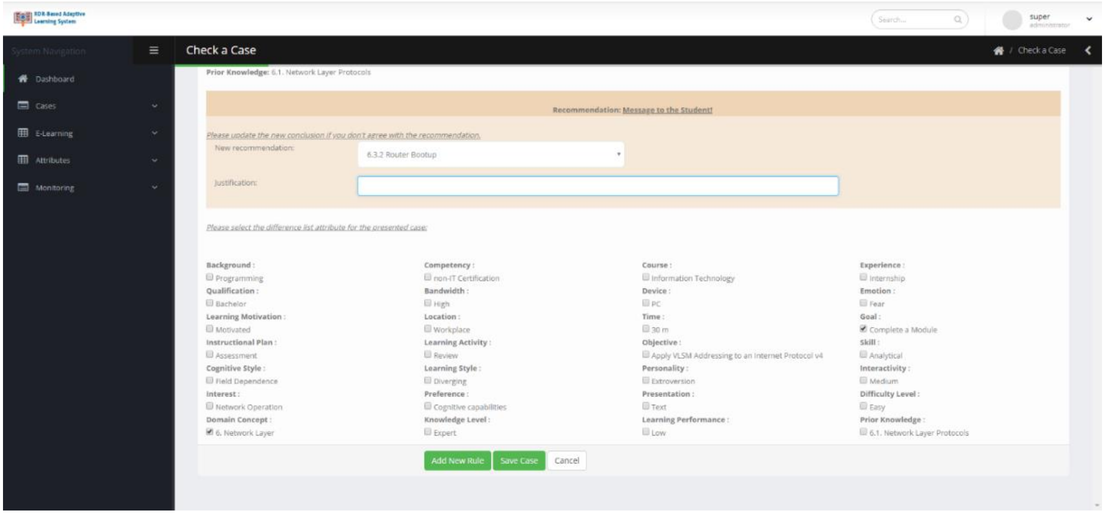


Figure 7.4 Inference result for the first test case

In this stage, the first test case has been successfully evaluated which resulted in the first rule R1 of the knowledge base. It can be seen in Figure 7.5 that the knowledge base has one rule R1 with a certain condition and conclusion. The presented case was also stored in the system as a cornerstone case for R1. We processed the acquisition incrementally and monitored the performance of knowledge in the next phase.

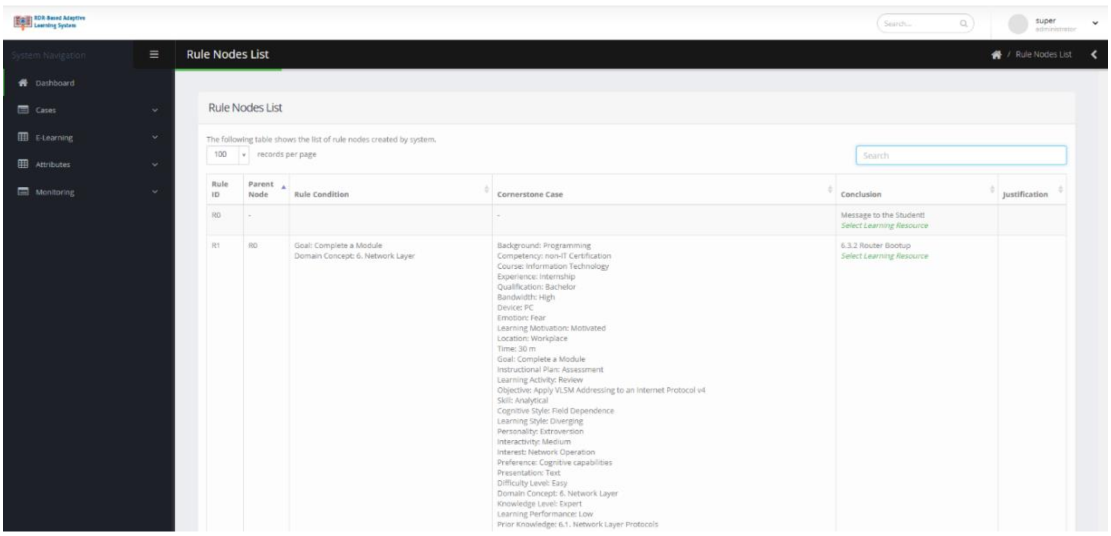


Figure 7.5 First rule created in the knowledge base

7.3.3 Phase 3: Analysis of the Evaluation

In this stage, the monitoring process of the knowledge base was undertaken. This is an automated process as received from log. The information generated from the knowledge base was processed and analysed according to the procedure described in section 5.2.3. For monitoring purposes, an analysis was undertaken based on correct performance. It means that a counter for each sequence (gap) which counts the correctly classified cases by that rule, was created. This counter was reset as soon as an incorrectly classified case was added to the knowledge base.

A total of 753 test cases were available for evaluation. The first sequence of ten cases resulted in an incorrect classification for all cases. As the KB did not have any rule before, those case were rejected, and the new rules were created every time an incorrectly classified case was found. In the next batch, the first correctly classified case was found after running 13 cases. As shown in Table 7.5, the system's output matched the expected output and no rule was created for case number 13. Hence, the monitoring module created the first correct gap for analysis. The value of the first gap only consisted of one correct case since the next case was classified as incorrect. Thus, the counter for the gap was reset and the system waited for the next correct case to be processed and analysed.

Further information of the results could be obtained for the monitoring module. The information was presented in the form of a table. As illustrated in Figure 7.6, the first gap for analysis has been created. This information contains statistical analysis as previously described in Chapter 5. In the next section, the results of our experiment are presented, and further discussions are conducted to understand and explain the role of analysing the knowledge base development.

Table 7.5 Sequence of the test cases

Case No.	Expected Output	System's Output	Accepted Conclusion	Build New Rule	Rule Name	Parent
1	6.3.2 Router Bootup	Message to Student!	No	Yes	R1	R0
2	4.5.2. Class Activity - Linked In!	6.3.2 Router Bootup	No	Yes	R2	R1
3	9.1.1. Transportation of Data	4.5.2. Class Activity - Linked In!	No	Yes	R3	R2
4	3.3.1. Data Encapsulation	6.3.2 Router Bootup	No	Yes	R4	R1
5	10.2.1. Common Application Layer Protocols	9.1.1. Transportation of Data	No	Yes	R5	R3
6	3.4.2. Quiz - 3	4.5.2. Class Activity - Linked In!	No	Yes	R6	R2
7	6.1.1. Network Layer in Communication	9.1.1. Transportation of Data	No	Yes	R7	R3
8	7.2.2. Testing and Verification	3.3.1. Data Encapsulation	No	Yes	R8	R4
9	4.1.1. Layer 2 Introduction	6.1.1. Network Layer in Communication	No	Yes	R9	R7
10	5.1.1. Ethernet Operation	3.4.2. Quiz - 3	No	Yes	R10	R6
11	9.1.1. Transportation of Data	4.1.1. Layer 2 Introduction	No	Yes	R11	R9
12	8.2.1. Structured Design	7.2.2. Testing and Verification	No	Yes	R12	R8
13	3.4.2. Quiz - 3	3.4.2. Quiz - 3	Yes	No	-	-
14	5.4.2. Class Activity - MAC and Choose	5.1.1. Ethernet Operation	No	Yes	R13	R10

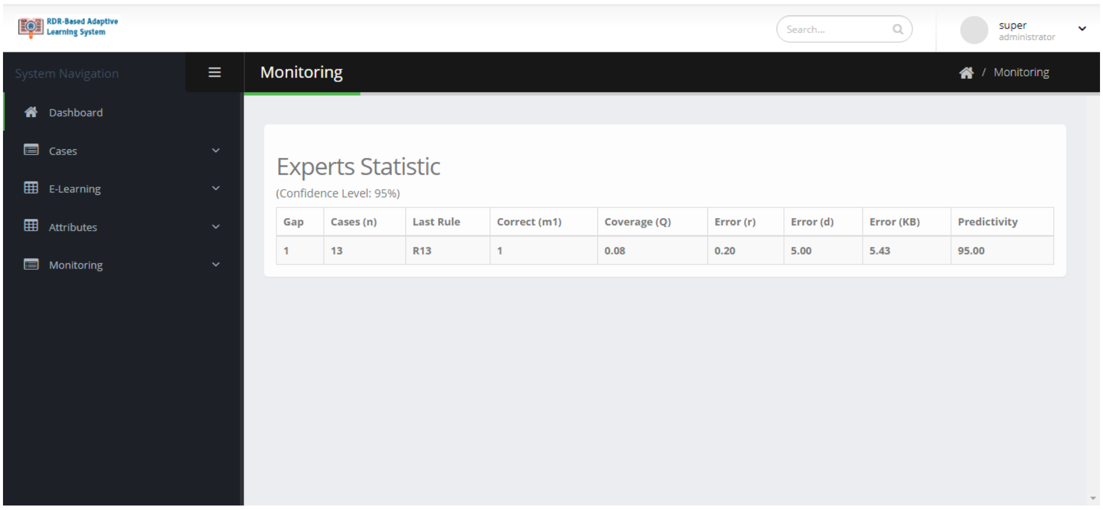


Figure 7.6 Monitoring module for expert

7.4 Experimental Results

The second experiment was run 161 times to evaluate test cases resulting in 25 gaps. Table 7.6 shows the actual results. Column two to four are the number of rules, number of cases, and correct cases in each gap, respectively. *Q* is the ratio of correct cases with total cases. This value is used for calculating the coverage. *Er* and *Ekb* represent the error of new rule r and error of the whole knowledge base. Last column *P* is the predictivity of the rule. With 95% confidence level, some *Er* values during early development stage indicate the high level of error as can be observed in gap 1, 4, 6, and 7. A high level of errors was also discovered in the middle development stage, as indicated in gaps 11, 13, and 17. The rest of the gaps, especially during the final development stage, indicated a low level of errors. The results of predictivity also show that the convergence of the knowledge base was established during the middle development stage (after gap 13), where the *P* values was more than 95%. It indicates that the evaluation of the test cases is unnecessary since the evaluation will not have any impact on the knowledge base.

Table 7.6 The results of 25 gaps of correct performance

Gap	R	n	m1	Q	Cov	Er	Ekb	P
1	R13	13	1	0.08	0.20	5.00%	5.43%	95.00%
2	R14	4	3	0.75	1.11	1.70%	6.70%	98.30%
3	R18	6	2	0.33	0.65	2.53%	3.81%	97.47%
4	R19	2	1	0.50	1.08	5.00%	10.00%	95.00%
5	R23	7	3	0.43	0.74	1.70%	2.97%	98.30%
6	R24	2	1	0.50	1.08	5.00%	10.00%	95.00%
7	R26	3	1	0.33	0.78	5.00%	7.53%	95.00%
8	R27	3	2	0.67	1.11	2.53%	7.53%	97.47%
9	R28	3	2	0.67	1.11	2.53%	7.53%	97.47%
10	R29	5	4	0.80	1.09	1.27%	6.27%	98.73%
11	R30	2	1	0.50	1.08	5.00%	10.00%	95.00%
12	R31	11	9	0.82	1.01	0.57%	3.10%	99.43%
13	R33	2	1	0.50	1.08	5.00%	10.00%	95.00%
14	R34	14	13	0.93	1.04	0.39%	5.39%	99.61%
15	R35	3	2	0.67	1.11	2.53%	7.53%	97.47%
16	R36	3	2	0.67	1.11	2.53%	7.53%	97.47%
17	R37	2	1	0.50	1.08	5.00%	10.00%	95.00%
18	R38	8	7	0.88	1.07	0.73%	5.73%	99.27%
19	R40	12	10	0.83	1.01	0.51%	3.04%	99.49%
20	R41	3	2	0.67	1.11	2.53%	7.53%	97.47%
21	R42	18	17	0.94	1.03	0.30%	5.30%	99.70%
22	R44	15	13	0.87	1.01	0.39%	2.93%	99.61%
23	R45	16	15	0.94	1.04	0.34%	5.34%	99.66%
24	R47	4	2	0.50	0.91	2.53%	5.06%	97.47%
25	R48	5	4	0.80	1.09	1.27%	6.27%	98.73%

7.5 Performance Evaluation of the Knowledge Base

In the second experiment, the system has evaluated 161 test cases and produced 48 rules to find 25 gaps of correct performance. As Figure 7.7 shows, the first sequence of 161 test cases was used to evaluate the initial KB. The rest of the 161 test cases were required for subsequent testing. Overall, 322 test cases were used for the second experiment. The initial testing was measured from the first cycle with an empty rule until the knowledge base constructed 48 rules. The subsequent testing was conducted after that, without any modification to the knowledge base. We used four parameters for performance measurement, with the accuracy as the basis and macro-average score of precision, recall, and F1 for the certainty.

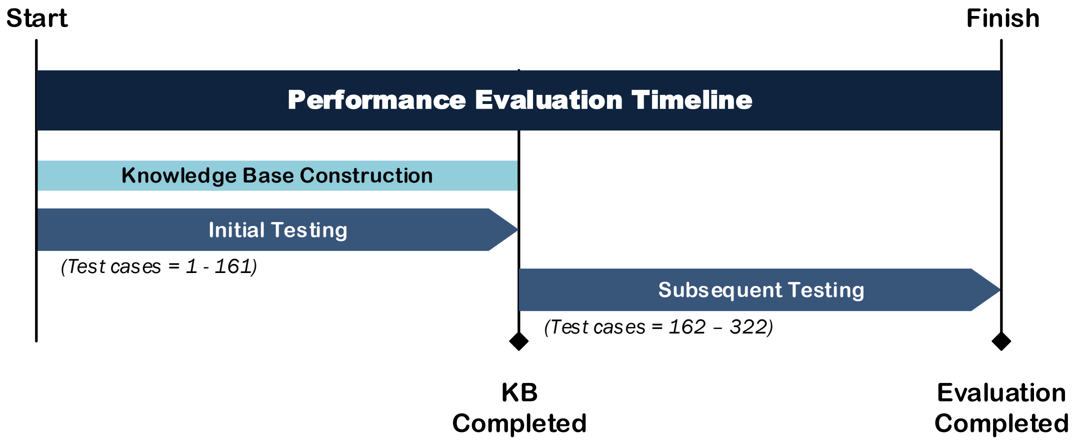


Figure 7.7 Performance evaluation plan

Table 7.7 illustrates the performance evaluation results of the KB. The initial testing showed that the overall accuracy of the KB was at 72.05%, while the macro-averages of precision, recall, and F1 were 65.97%, 65.59%, and 65.78%, respectively. The accuracy is simply a ratio of correctly predicted class, either positive or negative, to the entire class. The precision score is a proper measurement for this experiment, since the cost of false positive is high. However, the F1 score is needed for measurement to seek a balance between precision and recall. Furthermore, the accuracy score for subsequent testing is 20% higher than the initial testing. The precision and recall also show a higher score than the previous study, with 93.97% and 91.19%. The difference between both scores is only around 3%. The balance score of macro-averaged F1 is very similar to the overall accuracy, which is 92.56%.

Table 7.7 Performance evaluation results of the KB (%)

Testing	Accuracy	Precision	Recall	F1
Initial	72.05	65.97	65.59	65.78
Subsequent	92.55	93.97	91.19	92.56

Note that the number of predicted classes for this experiment is higher, with 29 classes, compared to 17 classes in the first experiment. With only 48 rules created, the constructed KB has shown a maturity to predict 29 classes in the test cases. The macro-averaged F1 score for initial testing in this experiment is also higher compared to the score of previous study, which was only 47.66%. The difference between the F1 score in both experiments is around 10%. Overall, the performance results from two experiments were fairly consistent since they used the same system but with a different number of predicted classes. Although the F1 score in the first experiment was poor, the second experiment with more extended acquisition could give the highest result.

7.6 Chapter Summary

The second case study was conducted to observe the statistical output in the advanced development stage. Thus, the experiment was longer, with 162 iterations to evaluate test cases. The statistical results have shown that the values for both predictivity and coverage were high. It shows that the F1 score for initial testing was medium with a score of only 65.78%. The performance for subsequent tests is 92.56% and should be considered sufficient for the domain. If the cost to keep an expert is not an issue, that the knowledge base can be improved by conducting more acquisition.

8 DISCUSSION

This chapter presents the discussion and analysis of the experiments in this study. In general, there are two case studies to determine whether the developed KBS could be evaluated immediately as the knowledge grows. Therefore in Section 8.1, the experimental results from two case studies were discussed and compared. Then, the performances were measured and analysed in Section 8.2. The validation results are presented in Section 8.3.

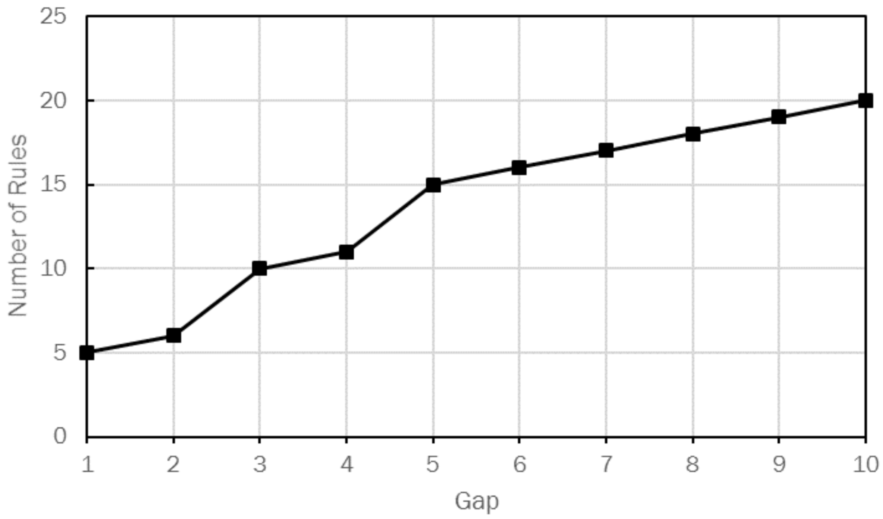
8.1 Discussions of the Experimental Results

Prior to this study, it was difficult to evaluate a KBS because of the cost and unavailability of experts. The problem for KBS evaluation process is that it needed to readily available and should demonstrate actual results for real applications for experts. Thus, the developer normally tested using available expert in the organisation and stored the database of cases. This database is used to compare the performance as typically used in machine learning. However, it is very difficult to use on a wide range of real world systems. Another major approach to evaluate was using other KBS as source of expertise. This problem is discussed in detail in (Compton, Preston et al. 1995). In this approach, the conclusions from simulated experts are used to evaluate the new KBS. Clearly any study in KBS development will use experts with a specific problem solving method to evaluate the system at the end of the KBS development. This research proposed evaluation by expert during knowledge acquisition process. The evaluation was taken to measure that the performance of the KB was acceptable during initial development. After the initial KB was built and initial testing result recorded, the knowledge acquisition process was halted, and no new rules were added to the knowledge base. This is to prevent knowledge modification in the KB.

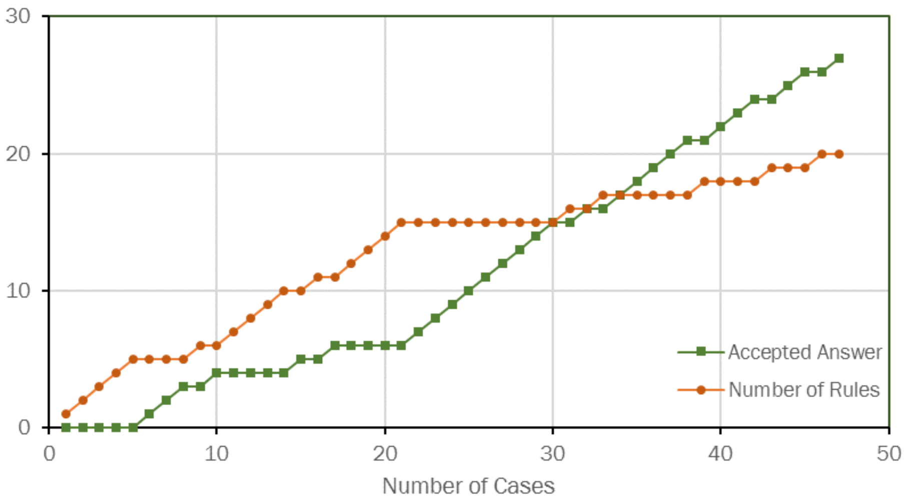
In order to validate the theoretical results presented in Chapter 5, two experiments were conducted to evaluate the adaptive e-learning system. The simulations correspond to two distinct learning case studies. An incremental-based AES called INCAES was utilised for monitoring the development of the knowledge base. The emphasis of the experiment is on developing the RDR KB and gauging its effectiveness by monitoring the knowledge acquisition process. The experiment ensures that the instructional knowledge base used to guide the dynamic learning is evaluated as it is constructed. The monitoring process is a key contribution of this thesis and it is underpinned by the following parameters in the knowledge acquisition process: the coverage, potential error, and predictivity of newly added rules. These parameters assess both the generalisability and accuracy of the newly added rules in the KB development.

The investigation of our analysis is based on the correct performance of the knowledge base. This is in line with the idea of RDR that only stores the rules with valid cases after validating incorrect cases. The first experiment has resulted in 392 available test cases to construct the knowledge base. However, only 48 test cases and 20 rules were required to find 10 gaps in the KB. Among the 48 test cases, 27 cases were correctly classified by the system, while the rest were incorrectly classified. The distribution of the rules and gaps in the KB is presented in Figure 8.1 (A). There is a clear trend of increasing rules as the gap grows, from 5 rules when the first gap found to 20 rules in the last gap. The number of rules was tripled in the first five gaps. This indicates the high coverage during the early growth of knowledge base. Then, the number of rules showed linear convergence by only increasing once during gap 6 to 10.

Even though the number of rules in RDR neither affect nor reflect the performance of the KB (Kang, Gambetta & Compton 1996), it might reveal the trend of acceptable answers of the knowledge base. As illustrated in Figure 8.1 (B), the accepted answers rise gradually while the number of rules show a steady growth. Figure 8.1 (B) also revealed that the KB was learning along with the increasing of experience (number of cases). After the knowledge base grew to about 15 rules in 30 cases, it indicated that the KB had reached its peak and was capable of answering correctly in response to the presented cases. It is an indication that the KB has converged and learned important information about the characteristics of the domain knowledge.



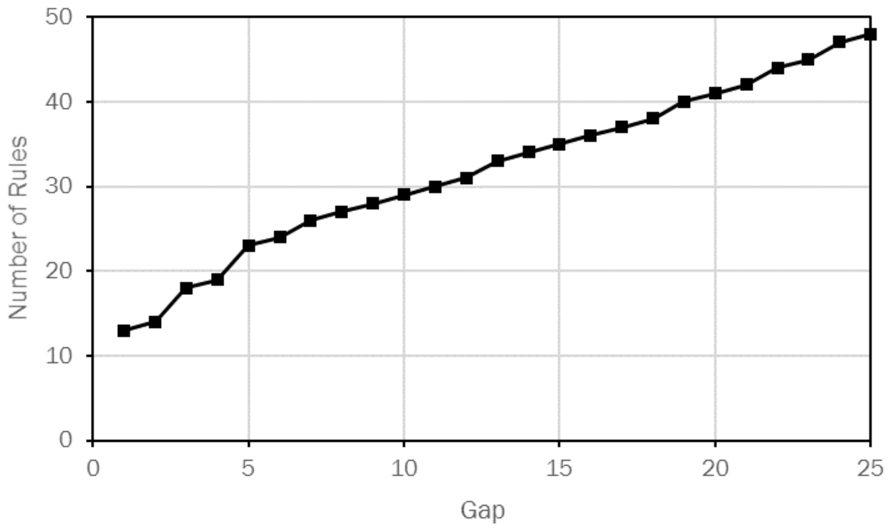
(A)



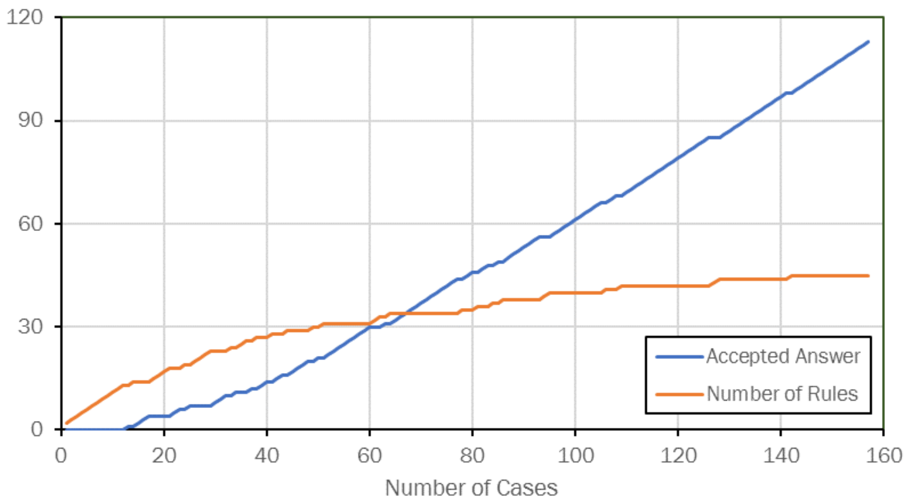
(B)

Figure 8.1 The distribution of rules (A) and accuracy of initial KB (B) for Case Study 1

In general, the second experiment needed 161 cases and created 48 new rules to find the 25 gaps of correct performance in the KB. To understand a trend of the gaps, a change of the number of rules in each gap and correctly classified cases over time is displayed in Figure 8.2 (A) and (B), respectively. As Figure 8.2 (A) shows, the first gap needed 13 rules, and a sharp increase in rules has shown until the fifth gap. After the fifth gap, the rules have grown at a steady, linear rate until the last 25 gaps. Furthermore, Figure 8.2 (B) shows the comparison of rule growth with the acceptable answer from the system. During the first early stage of the knowledge base construction, 14 test cases have failed to satisfy due to the empty knowledge in the KB. However, Figure 8.2 (B) shows that the number of accuracies show a linear rate in the advanced development stage. After evaluating the 60 test cases, it is discovered that the knowledge base already contains enough knowledge for the rest of the cases. The number of correctly classified cases had passed the number of false cases.



(A)



(B)

Figure 8.2 The distribution of rules (A) and accuracy of initial KB (B) for Case Study 2

8.2 Performance of Knowledge Base

For the first case study, the scenario was to observe the statistical output in the early development stage. The results have shown that the statistical output is low in predictivity but high in coverage. This is an indication that the rule is too general for the domain. The rule may cover the required condition, but the conclusion is not appropriate. It is further confirmed by the experiment, which showed in Table 8.1 that the performance of the system was poor during initial testing (F1 score=47.66%). Nevertheless, subsequent testing has shown that the performance of knowledge base was acceptable with F1 = 83.4%. This is not the greatest score for the system prediction. However, if the project manager should take a rapid decision for their project, they could take the decision early without running a longer knowledge acquisition. In this experiment, we obtained 10 gaps of correct performance. For a quicker decision, we suggested to reduce this number by monitoring 5–7 gaps only. For a greater result in the early development stage, they should expect high predictivity and high coverage from the statistical results.

For the second case study, the scenario was to observe the statistical output in the advanced development stage. Thus, the experiment was longer, with 162 iterations to evaluate test cases. The statistical results have shown that the values for both predictivity and coverage were high. According to the decision table above, this is an indication of appropriate rule set for the given domain. The indication is the high convergence at a later stage that shows the slow linear convergence. If we analysed the performance results, Table 8.1 shows that the F1 score for initial testing was medium with a score of only 65.78%. We believe that with 161 test cases and 48 rules, the initial testing could perform better. However, the performance for subsequent tests is 92.56% and should be considered sufficient for the domain. If the cost to keep an expert is not an issue, that the knowledge base can be improved by conducting more acquisition.

Table 8.1 Performances Comparison

	Case Study 1		Case Study 2	
	Initial	Subsequent	Initial	Subsequent
Accuracy	56.25	83.33	72.05	92.55
Precision	46.18	80.56	65.97	93.97
Recall	49.24	86.44	65.59	91.19
F1 Score	47.66	83.4	65.7	92.56

8.3 Validation Results

As presented earlier, there are several approaches of validation methodologies for knowledge-based systems. The question raised is how to decide which one to use. It is unlikely that one general method will suit for all domains, rather some validations are only suitable for certain domains. In this research, the KBS was developed and applied incrementally using RDR. This approach is commonly used to maintain the KB in routine use. Since the system continuously developing as the knowledge changes, validating such system is a really challenging task. In this research, the dynamic evaluation method was applied to validate the knowledge base in a realtime system. The definitive step in realtime validation is to employ statistical analysis module. This would not only save time but effort to analyse the continuation of the project.

Through the two experiments, we have confirmed the theoretical results on the topic of knowledge base evaluation process. As discussed in Chapter 2, there are three pieces of information that can be used to evaluate the success of the knowledge acquisition process. The first is the quality of the expertise that is reflected by estimating the predictivity. Secondly, the progress of the knowledge acquisition process. This information is obtained from the coverage value. During the early stage, the coverage of newly added rules is relatively high. When the coverage values are constant, the accuracy of the knowledge base tends to be acceptable. Lastly, whether or not the knowledge acquisition methodology is effective for the domain. Taking a decision will depend on the size of the domain and the available time resources that can be afforded towards it. Furthermore, Table 8.2 shows the decision table for guiding the KBS project of adaptive learning development. This gives insights into whether the expertise has been achieved and the knowledge acquisition project should be terminated.

Through the two experiments, we have analysed the effort required to develop the knowledge-based system project. The results have suggested that the learner model and the way materials are organised in domain model, already lends itself to dynamic e-learning domain. Specifically the description of the case from learner model, the learning pathway taken by learner, and the sequencing of the materials enables easier construction of the knowledge base. Although KBSs have reached the maturity stage in their development, there is still room for improvement in their evaluation stage. In this thesis, we supported the idea to integrate the development into the evaluation stage. The experiments have suggested that one of the interesting methods in this area is by monitoring knowledge acquisition during project development. We have shown here that combining the incremental RDR with statistical monitoring will handle both, development and evaluation tasks. With respect to the necessity of building an initial system and then conducting an evaluation, we have suggested that the statistical monitoring may provide sufficient information for the sustainability of the project development.

Table 8.2 Decision table used for analysing the knowledge acquisition monitor in adaptive learning (Adapted from (Beydoun & Hoffmann 2013, p. 245))

Parameters	Early Development Stage	Advanced Development Stage	Comments
Low predictivity + Low coverage	Overgeneralised rule and low accuracy	Overgeneralised rule and high accuracy	Linear convergence compounded with overgeneralised rule
Low predictivity + High coverage	Overgeneralised rules and average accuracy	Overspecialised rule and high accuracy	High coverage at a later stage may indicate slow linear convergence
High predictivity + High coverage	Appropriate rule set	Appropriate rule set and high accuracy	High coverage at a later stage may indicate fast linear convergence that applies when domain is large
High predictivity + Low coverage	Overspecialised rule / high accuracy	Appropriate rule set with high accuracy	Low coverage is expected at later knowledge acquisition stages when we have logarithmic convergence

9 CONCLUSION

This final chapter concludes the thesis by summarising and reflecting its main contribution. The chapter is organised as follows. Section 9.1 summarises the thesis. Section 9.2 discusses several approaches that contribute to the goal of the thesis. Section 9.3 outlines the limitation of the research. The final Section 9.4 provides the possibility for future research and conclude the thesis with final remarks.

9.1 Research Summary

This thesis has investigated the use of a knowledge base to guide students in a dynamic e-learning environment. At the same time, the thesis sought to contribute to one of the fundamental issues in evaluating knowledge-based systems: when is the knowledge base is ready to deliver with valid knowledge? To examine this issue, we focused on integrating the development phase with the testing phase of the knowledge bases developed. Based on our investigation, we have learnt that it is possible to provide a ready to use knowledge base without a separate testing phase by monitoring tasks during the knowledge acquisition process. Of course, since we have focussed on the knowledge acquisition in the context of dynamic e-learning, the results primarily uncovered the feasibility of the approach in the dynamic e-learning domain. This research tailored the dynamic evaluation framework and utilised the RDR technology as an incremental knowledge acquisition methodology with additional statistical analysis to estimate the effectiveness of the KBS in the dynamic e-learning domain. Furthermore, to demonstrate the proof of concept, an adaptive e-learning expert system was developed for suggesting learning resources to students by analysing their attributes. The process of constructing the knowledge base was validated through two case studies.

A design science research method was adopted to achieve the aims of the thesis. It consists of four phases, with problem identification in the first phase. Our investigation was initiated by studying relevant research related to the knowledge-based system, with more focus on the knowledge acquisition, evaluation, and validation processes. Despite its advantages, KBSs suffer from major drawbacks in a long evaluation process. This phase is critical for the further development of a KBS. Thus, this research focused on a dynamic evaluation approach by monitoring the knowledge acquisition process in RDR during the KB development process. This study was needed to better understand the mechanism underlying the approach as it has never been applied in practise. In particular, we evaluated the concept under two case studies related to adaptive e-learning systems.

Therefore, the second phase of the research explores the adaptive e-learning system, which includes modelling the learner model. This activity was described in Chapter 4. This phase was aimed to examine the main characteristics that a learner should have to assist them during the adaptation process. The important issue discussed in this phase is how a learner and their elements can be used to provide them with useful learning information that is useful for the adaptation process. The elements or attributes of the learner were extracted from various sources of knowledge. Then, the attributes and corresponding values were modelled for further case studies.

The next phase was the design and development of the main artefact, which was an instantiation to solve the problem as identified in the first phase. The instantiation is realised as a knowledge-based expert system that supports adaptive learning. An incremental knowledge acquisition RDR was used to construct the knowledge base in the system. There were also two requirements to be filled for the development process. First, the system should facilitate knowledge extraction and structuring of the knowledge from sources. All the KB rules were created automatically from errors that were found during knowledge acquisition process. However, an expert was still needed to perform manual refinement of the test cases. The second requirement is that the system should facilitate the monitoring of the knowledge acquisition process. The monitoring module was an add-on that was built to measure the performance of the KB directly during the knowledge acquisition process. The description of these phases was presented in Chapter 5.

The last phase of the research was focused on testing and evaluating the concept. We described the evaluation process and its performance comparison in Chapter 6 and 7, respectively. The first case study was applied on the Web Programming subject for undergraduate students in the IT/CS field. The KBS stored the learning materials which were tailored with the test case entered into the system. The test case included a pair of attributes and values that represented a learner in an adaptive learning environment. The evaluation was conducted by monitoring the progress of the first ten gaps of the knowledge acquisition process. Following the evaluation of the initial knowledge base, the experiment was conducted using the following test case to evaluate the performance of the final KB. The second case study used a similar approach and procedures. However, the knowledge base was instantiated with the domain knowledge from a Networking Essentials subject for undergraduate students. Different from the first case study, the decision to stop testing in the second case study was taken after a delay, by monitoring 25 gaps of correct performance. The results showed that the KB had very high accuracy and was ready to deliver.

9.2 Contribution

The main goal and overview of this thesis has been presented in the previous section. Several novel approaches have been demonstrated in this thesis which allowed us to achieve the main objectives. The findings from this study have made several contributions to the current literature. In particular, the contribution of the thesis could be viewed from two perspectives, theoretical contribution and practical contribution.

On the theoretical contribution side, this thesis has contributed to the body of knowledge by introducing a novel approach to construct a knowledge-based system that supports maintenance and evaluation immediately in one phase. Prior to this study, researchers had suggested to embed the validation into development in a software development lifecycle. This approach proved effective in the dynamic e-learning domain, and it needs further validation in other domains. However, the thesis makes a very significant finding about the feasibility a knowledge driven approach for dynamic e-learning.

For the knowledge acquisition community, concurrent evaluation and construction was difficult to support when building a KBS as it has a different lifecycle. The findings of the study support the view that the development of a KBS should be distinguished from conventional software (Batarseh & Gonzalez 2015), in that there are process models that are specific to KBS development and should be explored. Rather than validating a KBS after the development process, we exploited the knowledge acquisition process to analyse the performance of the knowledge base. This would allow experts to assess the whole KBS and take decisions based on the knowledge acquisition process.

As the knowledge base grows, our monitoring module analyses the performance of the KBS based on correct performance and reports the potential error of knowledge base to the experts. The statistical analysis successfully facilitates the experts to have a better understand of the domain knowledge and expertise. This process could reduce the cost of time of experts to maintain the knowledge acquisition process and take rapid decisions on whether to keep the knowledge base or not. Our research has also shown that the approach did not require the comparison of the knowledge base against external data source. As a result, this process would minimise considerable amount of data needed for evaluation.

The comparison of the performance analysis during knowledge acquisition process and after building the initial KB has been presented. This evaluation helps to understand the real performance of our analysis by measuring the value of precision, recall, and F-measure. The results have shown that the quality of the KB was acceptable and in line with the growth of the analysis gap from the KA process. Overall, our monitoring module helps the experts in minimising the cost of the evaluation process, while still keeping the accuracy of knowledge base at an acceptable level

On the practical contribution side, the thesis contributes to the adaptation of e-learning systems. It first presents a new learner model for adaptive e-learning systems based on recent works. We analysed 145 papers from reputable sources and extracted the concept from them. Besides renewing interest in the current literature, this process also helps to understand the characteristics of the students in adaptive learning environments that are constantly changing over time. The results suggested 28 attributes for students in AES. Furthermore, we presented an incremental based approach to suggest to educational experts to personalise student's learning paths based on cases and to store the rules in the knowledge base. This is a new approach as most of works designated the adaptation rules ahead for all the scenarios. Our system allows experts to define scenario on a case by case basis. The system does not allow direct rule creation nor correction to the existing rules as to maintain the integrity of the existing knowledge in the KB. All rules will be created automatically from errors that are found during knowledge acquisition process instead. The experts only handle manual refinement to distinguish the presented case with previously seen case.

9.3 Limitations

Since the main focus of this thesis is the process to construct a knowledge base and rapidly evaluate its performance, the following objectives were not pursued as they were outside the scope of the research:

1. During the generation phase of the test cases, this thesis will not provide a mechanism to validate the selection of results. The generation results of the pairwise testing will be used as created.
2. The thesis does not pass the responsibility to evaluate the usability of the developed system to the students. The evaluation process will be conducted by the experts. They will examine the knowledge base of the system and conclude the development status of the project. However, we demonstrate the potential use of the system to provide learning adaptation to students.

9.4 Recommendation for Future Research and Concluding Remarks

Based on the previous discussions and contributions of this study, some recommendations are suggested for future research. First, this study chooses pair wise testing as the method for generating test cases. Future research might investigate the mechanism to validate the test case and the effect of test cases sequencing with the speed of knowledge convergence. The validation of the test cases might ensure that the test cases have represented the domain. Then, the sequence of the test cases should be investigated since RDR builds the KB based on it. In addition, this approach proved effective in the dynamic e-learning domain. However, it needs further validation in other domains.

Secondly, during the evaluation process, this study suggested some numbers to obtain the gaps for further observation. This issue is vital to formalise the limit for early and advanced stages in project development. Therefore, future research opportunity is open to investigate whether the knowledge acquisition is in early or late stages.

Third suggestion is related to performance measurement. In this study, we used macro-average F1 for evaluating the performance. However, there are no guidelines as to the basis measurement for the evaluation. Future research is necessary to investigate the conditions in which the measurement indicates bad or good results. Fourthly, future studies might focus on investigating how far the costs of keeping experts could be reduced during system development projects. Since the methodology emerges from the problem of reducing the cost of maintaining an information system project, future investigating on how the model can support the project by providing a rapid decision on the knowledge base is very useful.

The last recommendation is related to adaptive e-learning as the domain of the study. This research has proposed a method for evaluating the running development of a knowledge base. Even though the purpose is for the decision making for project managers, the future evaluation of the work should include end user testing. After all, this system is intended for students to assist them in their learning paths. Furthermore, research work can be conducted by involving students and analysing the system recommendations to their preferences. If such research work can be done, then more contributions will be achieved in this field.

REFERENCES

Abdullah, U., Ligęza, A. & Zafar, K. 2017, 'Performance Evaluation of Rule-Based Expert Systems: An Example from Medical Billing Domain', *Expert Systems*.

Abech, M., da Costa, C.A., Barbosa, J.L.V., Rigo, S.J. & da Rosa Righi, R. 2016, 'A Model for Learning Objects Adaptation in Light of Mobile and Context-Aware Computing', *Personal and Ubiquitous Computing*, vol. 20, no. 2, p. 167.

Acampora, G., Gaeta, M. & Loia, V. 2011, 'Hierarchical optimization of personalized experiences for e-Learning systems through evolutionary models', *Neural Computing and Applications*, vol. 20, no. 5, pp. 641-57.

Adelman, L. & Riedel, S.L. 1997, *Handbook for Evaluating Knowledge-Based Systems: Conceptual Framework and Compendium of Methods*, Springer Publishing Company, Incorporated.

Agarwal, R. & Tanniru, M. 1990, 'Systems development life-cycle for expert systems', *Knowledge-Based Systems*, vol. 3, no. 3, pp. 170-80.

Ali, A., Abd Razak, S., Othman, S.H., Mohammed, A. & Saeed, F. 2017, 'A Metamodel for Mobile Forensics Investigation Domain', *PLOS ONE*, vol. 12, no. 4, pp. 1-32.

Ammar-Khodja, S., Perry, N. & Bernard, A. 2008, 'Processing Knowledge to Support Knowledge-Based Engineering Systems Specification', *Concurrent Engineering*, vol. 16, no. 1, pp. 89-101.

Anam, S., Kim, Y.S., Kang, B.H. & Liu, Q. 2015, 'Schema Mapping Using Hybrid Ripple-Down Rules', pp. 17-26.

Atif, Y., Benlamri, R. & Berri, J. 2003, 'Learning Objects Based Framework for Self-Adaptive Learning', *Education and Information Technologies*, vol. 8, no. 4, pp. 345-68.

Barker, V.E. & O'Connor, D.E. 1989, 'Expert systems for configuration at Digital: XCON and beyond', *Communications of the ACM*, vol. 32, no. 3, pp. 298-318.

Barr, V. 1999, 'Applications of rule-base coverage measures to expert system evaluation', *Knowledge-Based Systems*, vol. 12, no. 1, pp. 27-35.

Batarseh, F.A. & Gonzalez, A.J. 2013, 'Incremental lifecycle validation of knowledge-based systems through CommonKADS', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 3, pp. 643-54.

Batarseh, F.A. & Gonzalez, A.J. 2015, 'Validation of knowledge-based systems: a reassessment of the field', *Artificial Intelligence Review*, vol. 43, no. 4, pp. 485-500.

Bekmann, J. & Hoffmann, A. 2005, 'Improved knowledge acquisition for high-performance heuristic search', vol. 19, LAWRENCE ERLBAUM ASSOCIATES

LTD, p. 41.

Benlamri, R. & Zhang, X. 2014, 'Context-aware recommender for mobile learners', *Human-centric Computing and Information Sciences*, vol. 4, no. 1, pp. 1-34.

Beydoun, G. & Hoffmann, A. 1997a, 'Acquisition of search knowledge', *Knowledge Acquisition, Modeling and Management*, pp. 1-16.

Beydoun, G. & Hoffmann, A. 1997b, 'NRDR for the Acquisition of Search Knowledge', Springer, pp. 177-86.

Beydoun, G. & Hoffmann, A. 1998, 'Simultaneous modelling and knowledge acquisition using NRDR', in H.-Y. Lee & H. Motoda (eds), *PRICAI'98: Topics in Artificial Intelligence: 5th Pacific Rim International Conference on Artificial Intelligence Singapore, November 22–27, 1998 Proceedings*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 83-95.

Beydoun, G. & Hoffmann, A. 2000a, 'Incremental acquisition of search knowledge', *International Journal of Human-Computer Studies*, vol. 52, no. 3, pp. 493-530.

Beydoun, G. & Hoffmann, A. 2000b, 'Monitoring knowledge acquisition instead of evaluating knowledge bases', *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, pp. 41-57.

Beydoun, G. & Hoffmann, A. 2001, 'Theoretical basis for hierarchical incremental knowledge acquisition', *International Journal of Human-Computer Studies*, vol. 54, no. 3, pp. 407-52.

Beydoun, G. & Hoffmann, A. 2013, 'Dynamic evaluation of the development process of knowledge-based information systems', *Knowledge and information systems*, pp. 1-15.

Beydoun, G., Hoffmann, A., Breis, J.T.F., Bejar, R.M., Valencia-Garcia, R. & Aurum, A. 2005, 'Cooperative modelling evaluated', *International Journal of Cooperative Information Systems*, vol. 14, no. 01, pp. 45-71.

Beydoun, G., Low, G., Henderson-Sellers, B., Mouratidis, H., Gomez-Sanz, J.J., Pavon, J. & Gonzalez-Perez, C. 2009, 'FAML: a generic metamodel for MAS development', *IEEE Transactions on Software Engineering*, vol. 35, no. 6, pp. 841-63.

Beydoun, G., Low, G., Mouratidis, H. & Henderson-Sellers, B. 2009, 'A security-aware metamodel for multi-agent systems (MAS)', *Information and Software Technology*, vol. 51, no. 5, pp. 832-45.

Brinton, C.G., Rill, R., Ha, S., Chiang, M., Smith, R. & Ju, W. 2015, 'Individualization for Education at Scale: MIIC Design and Preliminary Evaluation', *IEEE Transactions on Learning Technologies*, vol. 8, no.

3.1, pp. 136-48.

Brown, E.J., Brailsford, T.J., Fisher, T. & Moore, A. 2009, 'Evaluating learning style personalization in adaptive systems: Quantitative methods and approaches', *IEEE*

Transactions on Learning Technologies, vol. 2, no. 1, pp. 10-22.

Brusilovski, P., Kobsa, A. & Nejdl, W. 2007, *The Adaptive Web: Methods and Strategies of Web Personalization*, vol. 4321, Springer Science & Business Media.

Brusilovsky, P. 2001, 'Adaptive Hypermedia', *User Modeling and User-Adapted Interaction*, vol. 11, no. 1-2, pp. 87-110.

Brusilovsky, P. & Millán, E. 2007, 'User Models for Adaptive Hypermedia and Adaptive Educational Systems', *The Adaptive Web. Lecture Notes in Computer Science*, vol. 4321, Springer, pp. 3-53.

Brusilovsky, P. & Peylo, C. 2003, 'Adaptive and intelligent web-based educational systems', *International Journal of Artificial Intelligence in Education*, vol. 13, pp. 159-72.

Buchanan, B.G. & Shortliffe, E.H. 1984, *Rule-based expert systems*, vol. 3, Addison-Wesley Reading, MA.

Burns, H.L. & Capps, C.G. 2013, 'Foundation of Intelligent Tutoring Systems: An Introduction', in M.C. Polson & J.J. Richardson (eds), *Foundations of Intelligent Tutoring Systems*, Psychology Press, p. 1.

Canales, A., Peña, A., Peredo, R., Sossa, H. & Gutiérrez, A. 2007, 'Adaptive and intelligent web based education system: Towards an integral architecture and framework', *Expert Systems with Applications*, vol. 33, no. 4, pp. 1076-89.

Cao, T.M. & Compton, P. 2005, 'A simulation framework for knowledge acquisition evaluation', Australian Computer Society, Inc., pp. 353-60.

Caraca-Valente, J.P., Morant, J.L., González, L. & Pazos, J. 1999, 'Knowledge-based systems' validation: when to stop running test cases', *International Journal of Human-Computer Studies*, vol. 51, no. 4, pp. 757-81.

Caravantes, A. & Galán, R. 2011, 'Generic educational knowledge representation for adaptive and cognitive systems', *Journal of Educational Technology & Society*, vol. 14, no. 3, p. 252.

Caro, M.F., Josyula, D.P., Cox, M.T. & Jiménez, J.A. 2014, 'Design and validation of a metamodel for metacognition support in artificial intelligent systems', *Biologically Inspired Cognitive Architectures*, vol. 9, pp. 82-104.

Chakraborty, S., Roy, D. & Basu, A. 2010, 'Development of knowledge based intelligent tutoring system', *Advanced Knowledge Based Systems: Model, Applications & Research*, vol. 1, pp. 74-100.

Chandrasekaran, B. 1986, 'Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design', *IEEE expert*, vol. 1, no. 3, pp. 23-30.

Chen, C.-M. 2008, 'Intelligent web-based learning system with personalized learning path guidance', *Computers & Education*, vol. 51, no. 2, pp. 787-814.

Chen, G.-D., Chang, C.-K. & Wang, C.-Y. 2008, 'Ubiquitous learning website: Scaffold learners by mobile devices with information-aware techniques', *Computers & education*, vol. 50, no. 1, pp. 77-90.

Chen, W., Niu, Z., Zhao, X. & Li, Y. 2014, 'A hybrid recommendation algorithm adapted in e-learning environments', *World Wide Web*, vol. 17, no. 2, pp. 271-84.

Chiang, F.-K., Zhu, G., Wang, Q., Cui, Z., Cai, S. & Yu, S. 2015, 'Research and trends in mobile learning from 1976 to 2013: A content analysis of patents in selected databases', *British Journal of Educational Technology*.

Chiang, F.-K., Zhu, G., Wang, Q., Cui, Z., Cai, S. & Yu, S. 2016, 'Research and trends in mobile learning from 1976 to 2013: A content analysis of patents in selected databases', *British Journal of Educational Technology*, vol. 47, no. 6, pp. 1006-19.

Cho, W.-C. & Richards, D. 2004, 'BayesTH-MCRDR algorithm for automatic classification of Web document', Springer, pp. 344-56.

Chrysafiadi, K. & Virvou, M. 2012, 'Evaluating the integration of fuzzy logic into the student model of a web-based learning environment', *Expert Systems with Applications*, vol. 39, no. 18, pp. 13127-34.

Chrysafiadi, K. & Virvou, M. 2013a, 'Dynamically Personalized E-Training in Computer Programming and the Language C', *IEEE Transactions on Education*, vol. 56, no. 4, pp. 385-92.

Chrysafiadi, K. & Virvou, M. 2013b, 'Student modeling approaches: A literature review for the last decade', *Expert Systems with Applications*, vol. 40, no. 11, pp. 4715-29.

Colace, F., De Santo, M., Di Stasi, D. & Casillo, M. 2016, 'An Adaptive Educational Hypermedia System for Supporting Students in their Traditional Learning Process in Computer Engineering Education', *International Journal of Engineering Education*, vol. 32, no. 4, pp. 1625-36.

Colace, F., De Santo, M. & Greco, L. 2014, 'E-learning and personalized learning path: A proposal based on the adaptive educational hypermedia system', *International Journal of Emerging Technologies in Learning (iJET)*, vol. 9, no. 2, pp. 9-16.

Compton, P., Edwards, G., Kang, B., Lazarus, L., Malor, R., Menzies, T., Preston, P., Srinivasan, A. & Sammut, C. 1991, 'Ripple Down Rules: Possibilities and Limitations', University of Calgary.

Compton, P., Edwards, G., Kang, B., Lazarus, L., Malor, R., Preston, P. & Srinivasan, A. 1992, 'Ripple Down Rules: Turning Knowledge Acquisition into Knowledge Maintenance', *Artificial Intelligence in Medicine*, vol. 4, no. 6, pp. 463-75.

Compton, P. & Jansen, R. 1990, 'A philosophical basis for knowledge acquisition', *Knowledge Acquisition*, vol. 2, no. 3, pp. 241-58.

Compton, P., Peters, L., Edwards, G. & Lavers, T.G. 2006, 'Experience with Ripple-Down Rules', *Knowledge-Based Systems*, vol. 19, no. 5, pp. 356-62.

Czerwonka, J. 2006, 'Pairwise Testing in Real World', vol. 200, Citeseer.

da Silva, J.C., Matelli, J.A. & Bazzo, E. 2014, 'Development of a knowledge-based system for cogeneration plant design: Verification, validation and lessons learned', *Knowledge-Based Systems*, vol. 67, pp. 230-43.

Daughenbaugh, R., Ensminger, D., Frederick, L. & Surry, D. 2002, *Does Personality Type Effect Online versus In-Class Course Satisfaction?*

Dazeley, R. & Kang, B.-H. 2003a, 'Weighted MCRDR: Deriving Information about Relationships between Classifications in MCRDR'.

Dazeley, R. & Kang, B.H. 2003b, 'Rated MCRDR: Finding non-Linear Relationships Between Classifications in MCRDR'.

de-Marcos, L., Martinez, J.-J., Gutiérrez, J.-A., Barchino, R., Hilera, J.-R., Otón, S. & Gutiérrez, J.-M. 2011, 'Genetic algorithms for courseware engineering', *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 7, pp. 3981-4004.

De Bra, P., Aroyo, L. & Cristea, A. 2004, 'Adaptive web-based educational hypermedia', *Web Dynamics*, Springer, pp. 387-410.

De Kok, D. 2010, 'Feature selection for fluency ranking', Association for Computational Linguistics, pp. 155-63.

Dogan, B. & Dikbıyık, E. 2016, 'OPCOMITS: Developing an adaptive and intelligent web based educational system based on concept map model', *Computer Applications in Engineering Education*, vol. 24, no. 5, pp. 676-91.

Drachsler, H., Hummel, H.G. & Koper, R. 2008, 'Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model', *International Journal of Learning Technology*, vol. 3, no. 4, pp. 404-23.

Eberle, P., Schwarzinger, C. & Stary, C. 2011, 'User modelling and cognitive user support: towards structured development', *Universal Access in the Information Society*, vol. 10, no. 3, pp. 275-93.

Economides, A.A. 2009, 'Adaptive context-aware pervasive and ubiquitous learning', *International Journal of Technology Enhanced Learning*, vol. 1, no. 3, pp. 169-92.

Education, G.I.f. 2018, *Education, work and Australian society in an AI world: A review of research literature and policy recommendations*, UNSW, Gonski Institute for Education.

Education, N.A.o. 2013, *Adaptive Educational Technologies: Tools for Learning, and for Learning about Learning*, ERIC, Washington, DC.

El-Korany, A., Rafea, A., Baraka, H. & Eid, S. 2000, 'A structured testing methodology for knowledge-based systems', Springer, pp. 427-36.

El Bouhdidi, J., Ghailani, M. & Fennan, A. 2013, 'A Probabilistic Approach for the Generation of Learning Sessions Tailored to the Learning Styles of Learners', *International Journal of Emerging Technologies in Learning (iJET)*, vol. 8, no. 6, pp. 42-9.

Elsom-Cook, M. 1993, 'Student modelling in intelligent tutoring systems', *Artificial Intelligence Review*, vol. 7, no. 3-4, pp. 227-40.

Engineering, N.A.o. 2014, *14 Grand Challenges for Engineering in the 21st Century*, National Academy of Engineering, viewed 18/09/2018, <http://www.engineeringchallenges.org/challenges.aspx>.

Feigenbaum, E.A., Buchanan, B.G. & Lederberg, J. 1970, 'On generality and problem solving: A case study using the DENDRAL program'.

Feldman, J., Monteserin, A. & Amandi, A. 2015, 'Automatic detection of learning styles: state of the art', *Artificial Intelligence Review*, vol. 44, no. 2, pp. 157-86.

Finlayson, A. & Compton, P. 2004, 'Incremental knowledge acquisition using RDR for soccer simulation', pp. 102-16.

Friesen, N., Fisher, S. & Roberts, A. 2004, 'CanCore Guidelines Version 2.0: Educational Category', *CanCore Guidelines for the Implementation of Learning Object Metadata (IEEE 1484.12.1-200)*.

Galgani, F., Compton, P. & Hoffmann, A. 2015, 'LEXA: Building knowledge bases for automatic legal citation classification', *Expert Systems with Applications*, vol. 42, no. 17–18, pp. 6391-407.

García, E., Romero, C., Ventura, S. & Castro, C.d. 2009, 'An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering', *User Modeling and User-Adapted Interaction*, vol. 19, no. 1, pp. 99-132.

Germanakos, P., Tsianos, N., Lekkas, Z., Mourlas, C. & Samaras, G. 2008, 'Capturing essential intrinsic user behaviour values for the design of comprehensive web-based personalized environments', *Computers in Human Behavior*, vol. 24, no. 4, pp. 1434-51.

Gevarter, W.B. 1984, 'Expert systems: Artificial intelligence applied', *Telematics and Informatics*, vol. 1, no. 3, pp. 239-51.

Gómez, S., Zervas, P., Sampson, D.G. & Fabregat, R. 2014, 'Context-aware adaptive and personalized mobile learning delivery supported by UoLmP', *Journal of King Saud University - Computer and Information Sciences*, vol. 26, no. 1, Supplement, pp. 47-61.

Graf, S. & Kinshuk 2008, 'Adaptivity and Personalization in Ubiquitous Learning Systems', in A. Holzinger (ed.), *HCI and Usability for Education and Work: 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008, Graz, Austria, November 20-21, 2008.*

Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 331-8.

Graf, S. & Liu, T.C. 2010, 'Analysis of learners' navigational behaviour and their learning styles in an online course', *Journal of Computer Assisted Learning*, vol. 26, no. 2, pp. 116-31.

Graf, S., Yang, G., Liu, T.-C. & Kinshuk 2009, 'Automatic, global and dynamic student modeling in a ubiquitous learning environment', *Knowledge Management & E-Learning: An International Journal (KM&EL)*, vol. 1, no. 1, pp. 18-35.

Granic, A. 2008, 'Intelligent Interfaces for Technology-Enhanced Learning', *Advances in Human Computer Interaction*, InTech.

Granić, A. & Nakić, J. 2010, 'Enhancing the learning experience: Preliminary framework for user individual differences', Springer, pp. 384-99.

Gregor, S. & Hart, D. 2010, *Information Systems Foundations Part Three : The Role of Design Science*, ANU Press, Canberra.

Grubišić, A., Stankov, S. & Žitko, B. 2015, 'Adaptive courseware: A literature review', *Journal of universal computer science*, vol. 21, no. 9, pp. 1168-209.

Hafidi, M. & Bensebaa, T. 2013, 'Design and evaluation of an adaptive and intelligent tutoring system by expert system', *Intelligent Decision Technologies*, vol. 7, no. 4, pp. 253-64.

Hamade, R.F., Moulianitis, V.C., D'Addonna, D. & Beydoun, G. 2010, 'A dimensional tolerancing knowledge management system using Nested Ripple Down Rules (NRDR)', *Engineering Applications of Artificial Intelligence*, vol. 23, no. 7, pp. 1140-8.

Harrington, R. & Loffredo, D.A. 2010, 'MBTI personality type and other factors that relate to preference for online versus face-to-face instruction', *The Internet and Higher Education*, vol. 13, no. 1, pp. 89-95.

Hartung, R. & Håkansson, A. 2007, 'Automated Testing for Knowledge Based Systems', in B. Apolloni, R.J. Howlett & L. Jain (eds), *Knowledge-Based Intelligent Information and Engineering Systems: 11th International Conference, KES 2007, XVII Italian Workshop on Neural Networks, Vietri sul Mare, Italy, September 12-14, 2007. Proceedings, Part I*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 270-8.

Hernández, Y., Sucar, L.E. & Arroyo-Figueroa, G. 2013, 'Affective modeling for an intelligent educational environment', *Intelligent and Adaptive Educational-Learning Systems*, Springer, pp. 3-24.

Hevner, A.R., March, S.T., Park, J. & Ram, S. 2004, 'Design Science in Information Systems Research', *MIS Quarterly*, vol. 28, no. 1, pp. 75-105.

Ho, V., Wobcke, W. & Compton, P. 2003, 'EMMA: an e-mail management assistant', IEEE, pp. 67-74.

Hoic-Bozic, N., Mornar, V. & Boticki, I. 2008, 'Introducing Adaptivity and Collaborative Support into a Web-Based LMS', *Computing and informatics*, vol. 27, no. 4, pp. 639-59.

Hussain, T., Wardman, R. & Shamey, R. 2005, 'A knowledge-based expert system for dyeing of cotton. Part 2: Testing and evaluation', *Coloration technology*, vol. 121, no. 2, pp. 59-63.

Hwang, G.-J., Kuo, F.-R., Yin, P.-Y. & Chuang, K.-H. 2010, 'A Heuristic Algorithm for planning personalized learning paths for context-aware ubiquitous learning', *Computers & Education*, vol. 54, no. 2, pp. 404-15.

IEEE 2002, *IEEE Standard for Learning Object Metadata*.

IEEE 2005, 'IEEE Standard for Software Verification and Validation', *IEEE Std 1012-2004 (Revision of IEEE Std 1012-1998)*, pp. 1-110.

IMS 2003, *IMS Simple Sequencing Best Practice and Implementation Guide*.

Jeong, H.-Y. 2013, 'A practical use of learning system using user preference in ubiquitous computing environment', *Multimedia tools and applications*, vol. 64, no. 2, pp. 491-504.

Jeremic, Z., Jovanovic, J. & Gasevic, D. 2009, 'Evaluating an intelligent tutoring system for design patterns: The DEPTHS experience', *Journal of Educational Technology & Society*, vol. 12, no. 2, p. 111.

Jeremić, Z., Jovanović, J. & Gašević, D. 2012, 'Student modeling and assessment in intelligent tutoring of software patterns', *Expert Systems with Applications*, vol. 39, no. 1, pp. 210-22.

Johannesson, P. & Perjons, E. 2014, *An Introduction to Design Science*, Springer.

Jovanović, J., Gašević, D. & Devedžić, V. 2006, 'Dynamic Assembly of Personalized Learning Content on the Semantic Web', in Y. Sure & J. Domingue (eds), *The Semantic Web: Research and Applications*, vol. 4011, Springer Berlin Heidelberg, pp. 545-59.

Jun-Ming, S., Shian-Shyong, T., Wei, W., Jui-Feng, W., Jin Tan David, Y. & Wen-Nung, T. 2006, 'Learning Portfolio Analysis and Mining for SCORM Compliant Environment', *Journal of Educational Technology & Society*, vol. 9, no. 1, pp. 262-75.

Kang, B., Compton, P. & Preston, P. 1995, 'Multiple classification ripple down rules: evaluation and possibilities'.

Kang, B.H., Gambetta, W. & Compton, P. 1996, 'Verification and validation with ripple-down rules', *International Journal of Human-Computer Studies*, vol. 44, no. 2, pp. 257-69.

Karampiperis, P., Lin, T., Sampson, D.G. & Kinshuk 2006, 'Adaptive cognitive-based selection of learning objects', *Innovations in Education and Teaching International*, vol. 43, no. 2, pp. 121-35.

Karampiperis, P. & Sampson, D. 2005, 'Adaptive learning resources sequencing in educational hypermedia systems', *Journal of Educational Technology & Society*, vol. 8, no. 4, pp. 128-47.

Kardan, A.A., Aziz, M. & Shahpasand, M. 2015, 'Adaptive systems: a content analysis on technical side for e-learning environments', *Artificial Intelligence Review*, vol. 44, no. 3, pp. 365-91.

Kelly, D. 2008, 'Adaptive Versus Learner Control in a Multiple Intelligence Learning Environment', *Journal of Educational Multimedia & Hypermedia*, vol. 17, no. 3, pp. 307-36.

Kim, D., Han, S.C., Lin, Y., Kang, B.H. & Lee, S. 2018, 'RDR-based knowledge based system to the failure detection in industrial cyber physical systems', *Knowledge-Based Systems*, vol. 150, pp. 1-13.

Kim, J., Lee, A. & Ryu, H. 2013, 'Personality and its effects on learning performance: Design guidelines for an adaptive e-learning system based on a user model', *International Journal of Industrial Ergonomics*, vol. 43, no. 5, pp. 450-61.

Kim, M. & Compton, P. 2004, 'Evolutionary document management and retrieval for specialized domains on the web', *International Journal of Human-Computer Studies*, vol. 60, no. 2, pp. 201-41.

Klašnja-Milićević, A., Vesin, B., Ivanović, M. & Budimac, Z. 2011, 'E-Learning personalization based on hybrid recommendation strategy and learning style identification', *Computers & Education*, vol. 56, no. 3, pp. 885-99.

Knauf, R., Gonzalez, A.J. & Abel, T. 2002, 'A Framework for Validation of Rule-Based Systems', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*.

Knauf, R., Tsuruta, S. & Gonzalez, A.J. 2007, 'Toward reducing human involvement in validation of knowledge-based systems', *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 1, pp. 120-31.

Kuhn, D.R., Kacker, R.N. & Lei, Y. 2013, *Introduction to Combinatorial Testing*, Chapman & Hall/CRC Press.

Kurilovas, E., Zilinskiene, I. & Dagiene, V. 2014, 'Recommending suitable learning paths according to learners' preferences: Experimental research results', *Computers in Human Behavior*.

Lamia, M. & Tayeb, L.M. 2013, 'Discovering Learner Styles in Adaptive e-Learning Hypermedia Systems', *Journal of Universal Computer Science*, vol. 19, no. 11, pp. 1522-42.

Latham, A., Crockett, K. & McLean, D. 2014, 'An adaptation algorithm for an intelligent natural language tutoring system', *Computers & Education*, vol. 71, pp. 97-110.

Lee, S. & Lee, K.C. 2012, 'Context-prediction performance by a dynamic Bayesian network: Emphasis on location prediction in ubiquitous decision support environment', *Expert Systems with Applications*, vol. 39, pp. 4908–14.

Legaspi, R., Sison, R., Fukui, K.-i. & Numao, M. 2008, 'Cluster-based predictive modeling to improve pedagogic reasoning', *Computers in Human Behavior*, vol. 24, no. 2, pp. 153-72.

Liao, S.-H. 2005, 'Expert system methodologies and applications—a decade review from 1995 to 2004', *Expert systems with applications*, vol. 28, no. 1, pp. 93-103.

Limongelli, C., Sciarrone, F., Temperini, M. & Vaste, G. 2009, 'Adaptive Learning with the LS-Plan System: A Field Evaluation', *IEEE Transactions on Learning Technologies*, vol. 2, no. 3, pp. 203-15.

Lin, J., Sun, G., Cui, T., Shen, J., Xu, D., Beydoun, G., Yu, P., Pritchard, D., Li, L. & Chen, S. 2019, 'From ideal to reality: segmentation, annotation, and recommendation, the vital trajectory of intelligent micro learning', *World Wide Web*, pp. 1-21.

Lin, J., Sun, G., Shen, J., Cui, T., Yu, P., Xu, D., Li, L. & Beydoun, G. 2019, 'Towards the Readiness of Learning Analytics Data for Micro Learning', Springer, pp. 66-76.

Lundsgaarde, H.P. 1987, 'Evaluating medical expert systems', *Social Science & Medicine*, vol. 24, no. 10, pp. 805-19.

Mahnane, L., Laskri, M.T. & Trigano, P. 2013, 'A model of adaptive e-learning hypermedia system based on thinking and learning styles', *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, no. 3, pp. 339-50.

Manouselis, N., Drachsler, H., Verbert, K. & Santos, O.C. 2014, *Recommender systems for technology enhanced learning: research trends and applications*, Springer Science & Business Media.

Marković, S., Jovanović, Z., Jovanović, N., Jevremović, A. & Popović, R. 2013, 'Adaptive distance learning and testing system', *Computer Applications in Engineering Education*, vol. 21, no. S1, pp. E2-E13.

Martins, C., Faria, L., De Carvalho, C.V. & Carrapatoso, E. 2008, 'User modeling in adaptive hypermedia educational systems', *Educational Technology & Society*, vol. 11, no. 1, pp. 194-207.

McCrae, R.R. & John, O.P. 1992, 'An introduction to the five-factor model and its applications', *Journal of personality*, vol. 60, no. 2, pp. 175-215.

McDermott, J. 1982, 'R1: A rule-based configurer of computer systems', *Artificial intelligence*, vol. 19, no. 1, pp. 39-88.

Medina-Medina, N., Molina-Ortiz, F. & García-Cabrera, L. 2011, 'Adaptation and user modeling in hypermedia learning environments using the SEM-HP model and the JSEM-HP tool', *Knowledge and Information Systems*, vol. 29, no. 3, pp. 629-56.

Menzies, T. 1998, 'Evaluation issues with critical success metrics'.

Mitchell, T.M. 1997, 'Machine Learning', McGraw-Hill, MA: Boston.

Muhammad, A., Shen, J., Beydoun, G. & Xu, D. 2018, 'SBAR: A Framework to Support Learning Path Adaptation in Mobile Learning'Springer Singapore, Singapore, pp. 655-65.

Muhammad, A., Zhou, Q., Beydoun, G., Xu, D. & Shen, J. 2016, 'Learning path adaptation in online learning systems', pp. 421-6.

Müller, C. & Kohlhase, M. 2009, 'Context-aware adaptation: A case study on mathematical notations', *Information Systems Management*, vol. 26, no. 3, pp. 215-30.

Myers, M.D. & Venable, J.R. 2014, 'A set of ethical principles for design science research in information systems', *Information & Management*, vol. 51, no. 6, pp. 801-9.

Nakic, J., Granic, A. & Glavinic, V. 2015, 'ANATOMY OF STUDENT MODELS IN ADAPTIVE LEARNING SYSTEMS: A SYSTEMATIC LITERATURE REVIEW OF INDIVIDUAL DIFFERENCES FROM 2001 TO 2013', *Journal of Educational Computing Research*, vol. 51, no. 4, pp. 459-89.

Nguyen, D.Q., Nguyen, D.Q., Pham, D.D. & Pham, S.B. 2016, 'A robust transformation-based learning approach using ripple down rules for part-of-speech tagging', *AI Communications*, vol. 29, no. 3, pp. 409-22.

Nguyen, D.Q., Nguyen, D.Q. & Pham, S.B. 2014, 'Ripple Down Rules for question answering', *Semantic Web*, no. Preprint, pp. 1-22.

Nick, M., Althoff, K.-D. & Tautz, C. 1999, 'Facilitating the practical evaluation of organizational memories using the goal-question-metric technique'.

O'Keefe, R.M. & O'Leary, D.E. 1993, 'Expert system verification and validation: a survey and tutorial', *Artificial Intelligence Review*, vol. 7, no. 1, pp. 3-42.

Onoyama, T. & Tsuruta, S. 2000, 'Validation method for intelligent systems', *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 12, no. 4, pp. 461-72.

Ouf, S., Abd Ellatif, M., Salama, S.E. & Helmy, Y. 2017, 'A proposed paradigm for smart learning environment based on semantic web', *Computers in Human Behavior*, vol. 72, pp. 796-818.

Özpolat, E. & Akar, G.B. 2009, 'Automatic detection of learning styles for an e-learning system', *Computers & Education*, vol. 53, no. 2, pp. 355-67.

Özyurt, Ö. & Özyurt, H. 2015, 'Learning style based individualized adaptive e-learning environments: Content analysis of the articles published from 2005 to 2014', *Computers in Human Behavior*, vol. 52, pp. 349-58.

Paneva, D. 2006, 'Ontology-based student modeling', *Proceedings of the Fourth CHIRON*, vol. 6.

Papadimitriou, A., Grigoriadou, M. & Gyftodimos, G. 2009, 'Interactive Problem Solving Support in the Adaptive Educational Hypermedia System MATHEMA', *IEEE Transactions on Learning Technologies*, vol. 2, no. 2, pp. 93-106.

Papanikolaou, K.A. 2015, 'Constructing Interpretative Views of Learners' Interaction Behavior in an Open Learner Model', *IEEE Transactions on Learning Technologies*, vol. 8, no. 2, pp. 201-14.

Papanikolaou, K.A., Grigoriadou, M., Kornilakis, H. & Magoulas, G. 2003, 'Personalizing the Interaction in a Web-based Educational Hypermedia System: the case of INSPIRE', *User Modeling and User-Adapted Interaction*, vol. 13, no. 3, pp. 213-67.

Peffers, K.E.N., Tuunanen, T., Rothenberger, M.A. & Chatterjee, S. 2007, 'A Design Science Research Methodology for Information Systems Research', *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-77.

Peña-Ayala, A. 2012, *Intelligent and Adaptive Educational-Learning Systems: Achievements and Trends*, Springer Publishing Company, Incorporated.

Peña-Ayala, A., Sossa, H. & Méndez, I. 2014, 'Activity theory as a framework for building adaptive e-learning systems: A case to provide empirical evidence', *Computers in Human Behavior*, vol. 30, pp. 131-45.

Peña Ayala, A. 2010, 'Acquisition, representation and management of user knowledge', *Expert Systems with Applications*, vol. 37, no. 3, pp. 2255-64.

Pernas, A.M., Diaz, A., Motz, R. & de Oliveira, J.P.M. 2012, 'Enriching adaptation in e-learning systems through a situation-aware ontology network', *Interactive Technology and Smart Education*, vol. 9, no. 2, pp. 60-73.

Plant, R. & Gamble, R. 2003, 'Methodologies for the development of knowledge-based systems, 1982–2002', *The Knowledge Engineering Review*, vol. 18, no. 1, pp. 47-81.

Popescu, E. 2009, 'Diagnosing students’ learning style in an educational hypermedia system', *Cognitive and emotional processes in Web-based education: Integrating human factors and personalization, advances in Web-based learning book series, IGI Global*, pp. 187-208.

Prayote, A. & Compton, P. 2006, 'Detecting anomalies and intruders', Springer, pp. 1084-8.

Premlatha, K.R. & Geetha, T.V. 2015, 'Learning content design and learner adaptation for adaptive e-learning environment: a survey', *Artificial Intelligence Review*, pp. 1-23.

Preston, P., Edwards, G. & Compton, P. 1994, 'A 2000 rule expert system without a knowledge engineer', pp. 17.1-.0.

Ramadan, Z., Compton, P., Preston, P., Le-Gia, T., Chellen, V., Mulholland, M., Hibbert, D., Haddad, P. & Kang, B. 1998, 'From multiple classification RDR to configuration RDR', Citeseer.

Ramsey, C.L. & Basili, V.R. 1989, 'An evaluation of expert systems for software engineering management', *IEEE Transactions on Software Engineering*, vol. 15, no. 6, pp. 747-59.

Rastegarmoghadam, M. & Ziarati, K. 2016, 'Improved modeling of intelligent tutoring systems using ant colony opt', *Education and Information Technologies*, pp. 1-21.

Reategui, E., Boff, E. & Campbell, J.A. 2008, 'Personalization in an interactive learning environment through a virtual character', *Computers & Education*, vol. 51, no. 2, pp. 530-44.

Rhee, C. & Rao, H.R. 2008, 'Evaluation of decision support systems', *Handbook on Decision Support Systems 2*, Springer, pp. 313-27.

Richards, D. 2009, 'Two decades of ripple down rules research', *The Knowledge Engineering Review*, vol. 24, no. 2, pp. 159-84.

Richards, D. & Compton, P. 1997, 'Combining formal concept analysis and ripple down rules to support the reuse of knowledge'.

Richards, D. & Compton, P. 1999, 'Revisiting Sisyphus I-an Incremental Approach to Resource Allocation Using Ripple-Down Rules', SRDG Publications, University of Calgary, pp. 7-.1.

Richards, D. & Malik, U. 2002, 'Multi-level Rule Discovery from Propositional Knowledge Bases', pp. 11-9.

Richards, D.C. 1998, 'The Reuse of Knowledge in Ripple Down Rule Knowledge Based Systems', University of New South Wales.

Romero, C. & Ventura, S. 2010, 'Educational Data Mining: A Review of the State of the Art', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 6, pp. 601-18.

Russell, S.J. & Norvig, P. 2016, *Artificial intelligence: a modern approach*, Malaysia; Pearson Education Limited.

Sajja, P.S. & Akerkar, R. 2010, 'Knowledge-Based Systems for Development', *Advanced Knowledge Based Systems: Model, Applications & Research*, vol. 1, pp. 1-11.

Salehi, M., Nakhai Kamalabadi, I. & Ghaznavi Ghoushchi, M.B. 2014, 'Personalized recommendation of learning material using sequential pattern mining and attribute based collaborative filtering', *Education and Information Technologies*, vol. 19, no. 4, pp. 713-35.

Salle, A.J.L. & Medsker, L.R. 1990, 'The expert system life cycle: what have we learned from software engineering?', paper presented to the *Proceedings of the 1990 ACM SIGBDP conference on Trends and directions in expert systems*, Orlando, Florida, USA.

Sampson, D., Karagiannidis, C. & Kinshuk 2002, 'Personalised Learning: Educational, Technological and Standardisation Perspective', *Digital Education Review*, no. 4, pp. 24-39.

Sanchez, J. 2016, 'A Review of Pair-wise Testing', *arXiv preprint arXiv:1606.00288*.

Sangineto, E., Capuano, N., Gaeta, M. & Micarelli, A. 2008, 'Adaptive course generation through learning styles representation', *Universal Access in the Information Society*, vol. 7, no. 1-2, pp. 1-23.

Santos Jr, E. & Dinh, H.T. 2008, 'On automatic knowledge validation for Bayesian knowledge bases', *Data Knowledge Engineering*, vol. 64, no. 1, pp. 218-41.

Scheffer, T. 1996, 'Algebraic foundation and improved methods of induction of ripple down rules'.

Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., de Velde, W.V. & Wielinga, B. 2000, *Knowledge engineering and management : the Commonkads methodology*, The MIT Press, Cambridge (Massachusetts).

Schreiber, G., Wielinga, B., de Hoog, R., Akkermans, H. & Van de Velde, W. 1994, 'CommonKADS: A comprehensive methodology for KBS development', *IEEE expert*, vol. 9, no. 6, pp. 28-37.

Scott, E., Soria, A. & Campo, M. 2017, 'Adaptive 3D Virtual Learning Environments—A Review of the Literature', *IEEE Transactions on Learning Technologies*, vol. 10, no. 3, pp. 262-76.

Shi, L., Cristea, A.I., Foss, J.G., Al Qudah, D. & Qaffas, A. 2013, 'A social personalized adaptive e-learning environment: a case study in Topolor', *IADIS International Journal on WWW/Internet*, vol. 11, no. 3, pp. 1-17.

Shirazi, H. & Sammut, C. 2008, 'Acquiring control knowledge from examples using ripple-down rules and machine learning', *Iranian Journal of Science and Technology*, vol. 32, no. B3, p. 295.

Shute, V. & Towle, B. 2003, 'Adaptive E-Learning', *Educational Psychologist*, vol. 38, no. 2, pp. 105-14.

Software, I.S. 2016, *PictMaster Project*, viewed 10 September 2016.

Sokolova, M. & Lapalme, G. 2009, 'A systematic analysis of performance measures for classification tasks', *Information Processing & Management*, vol. 45, no. 4, pp. 427-37.

Sottilare, R.A., Graesser, A., Hu, X. & Goldberg, B. 2014, *Design Recommendations for Intelligent Tutoring Systems: Volume 2-Instructional Management*, vol. 2, US Army Research Laboratory.

Sottilare, R.A., Graesser, A., Hu, X. & Holden, H. 2013, *Design Recommendations for Intelligent Tutoring Systems: Volume 1-Learner Modeling*, vol. 1, US Army Research Laboratory.

Su, J.-M., Tseng, S.-S., Lin, H.-Y. & Chen, C.-H. 2011, 'A personalized learning content adaptation mechanism to meet diverse user needs in mobile learning environments', *User Modeling and User-Adapted Interaction*, vol. 21, no. 1-2, pp. 5-49.

Sun, G., Cui, T., Beydoun, G., Chen, S., Dong, F., Xu, D. & Shen, J. 2017, 'Towards massive data and sparse data in adaptive micro open educational resource recommendation: a study on semantic knowledge base construction and cold start problem', *Sustainability*, vol. 9, no. 6, p. 898.

Sun, G., Cui, T., Beydoun, G., Chen, S., Xu, D. & Shen, J. 2017, 'Organizing Online Computation for Adaptive Micro Open Education Resource Recommendation', *Advances in Web-Based Learning – ICWL 2017*, pp. 177-82.

Suryanto, H. & Compton, P. 2000, 'Learning Classification taxonomies from a classification knowledge based system'.

Tarpin-Bernard, F. & Habieb-Mammar, H. 2005, 'Modeling Elementary Cognitive Abilities for Adaptive Hypermedia Presentation', *User Modeling and User-Adapted Interaction*, vol. 15, no. 5, pp. 459-95.

Tsai, W.-T., Vishnuvajjala, R. & Zhang, D. 1999, 'Verification and validation of knowledge-based systems', *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, pp. 202-12.

UTS 2018a, *31270 Networking Essentials - Information Technology*, University of Technology Sydney, Sydney, viewed 18 April 2018, <http://handbook.uts.edu.au/subjects/details/31270.html>.

UTS 2018b, *31748 Programming on the Internet*, University of Technology Sydney, Sydney, viewed 5 April 2018, <http://handbook.uts.edu.au/2018/subjects/31748.html>.

Vaishnavi, V. & Kuechler, B. 2004, *Design Science Research in Information Systems*, 2, viewed July 2015, <http://desrist.org/desrist/content/design-science-research-in-information-systems.pdf>.

Van Asch, V. 2013, 'Macro-and micro-averaged evaluation measures [[basic draft]]'.

Vandewaetere, M., Desmet, P. & Clarebout, G. 2011, 'The contribution of learner characteristics in the development of computer-based adaptive learning environments', *Computers in Human Behavior*, vol. 27, no. 1, pp. 118-30.

Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I. & Duval, E. 2012, 'Context-Aware Recommender Systems for Learning: A Survey and Future Challenges', *IEEE Transactions on Learning Technologies*, vol. 5, no. 4, pp. 318-35.

Verginis, I., Gouli, E., Gogoulou, A. & Grigoriadou, M. 2011, 'Guiding Learners into

Reengagement through the SCALE Environment: An Empirical Study', *IEEE Transactions on Learning Technologies*, vol. 4, no. 3, pp. 275-90.

Vermesan, A. & Coenen, F. 2013, *Validation and Verification of Knowledge Based Systems: Theory, Tools and Practice*, Springer Science & Business Media.

Vesin, B., Klašnja-Milićević, A., Ivanović, M. & Budimac, Z. 2013, 'APPLYING RECOMMENDER SYSTEMS AND ADAPTIVE HYPERMEDIA FOR E-LEARNING PERSONALIZATION', *Computing & Informatics*, vol. 32, no. 3, pp. 629-59.

Wagner, W.P. 2017, 'Trends in expert system development: A longitudinal content analysis of over thirty years of expert system case studies', *Expert Systems with Applications*, vol. 76, pp. 85-96.

Wang, H.-C. & Huang, T.-H. 2011, 'Personalized e-learning environment for bioinformatics', *Interactive Learning Environments*, vol. 21, no. 1, pp. 18-38.

Weitzel, J.R. & Kerschberg, L. 1989, 'A system development methodology for knowledge-based systems', *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 3, pp. 598-605.

Wiedmann, T., Barbosa, J.L.V., Rigo, S.J. & Barbosa, D.N.F. 2016, 'RecSim: A Model for Learning Objects Recommendation using Similarity of Sessions', *Journal of Universal Computer Science*, vol. 22, no. 8, pp. 1175-200.

Wielinga, B.J., Schreiber, A.T. & Breuker, J.A. 1992, 'KADS: A modelling approach to knowledge engineering', *Knowledge acquisition*, vol. 4, no. 1, pp. 5-53.

Wieringa, R.J. 2014, *Design science methodology for information systems and software engineering*, Springer.

Xu, D., Huang, W.W., Wang, H. & Heales, J. 2014, 'Enhancing e-learning effectiveness using an intelligent agent-supported personalized virtual learning environment: An empirical investigation', *Information & Management*, vol. 51, no. 4, pp. 430-40.

Yang, F., Li, F.B. & Lau, R.H. 2010, 'An Open Model for Learning Path Construction', in X. Luo, M. Spaniol, L. Wang, Q. Li, W. Nejdl & W. Zhang (eds), *Advances in Web-Based Learning – ICWL 2010*, vol. 6483, Springer Berlin Heidelberg, pp. 318-28.

Yang, J., Huang, Z.X., Gao, Y.X. & Liu, H.T. 2014, 'Dynamic Learning Style Prediction Method Based on a Pattern Recognition Technique', *IEEE Transactions on Learning Technologies*, vol. 7, no. 2, pp. 165-77.

Yin, P.-Y., Chuang, K.-H. & Hwang, G.-J. 2014, 'Developing a context-aware ubiquitous learning system based on a hyper-heuristic approach by taking real-world constraints into account', *Universal Access in the Information Society*, pp. 1-14.