EDIT: An Educational Design Intelligence Tool for Supporting Design Decisions

Alaa Abdullah Bafail

A Thesis Submitted in Partial Fulfilment of the Requirements of Nottingham Trent University for the Degree of Doctor of Philosophy

September 2016

Acknowledgement

First and above all, I thank and praise Allah Almighty for giving me the strength and will to complete this research successfully and to write my PhD research under short period of time.

I would like to express my deepest thanks to my supervisor Dr. Jonathon Tepper, for taking me as a PhD student first and for his constant support, guidance and patience to accomplish the tasks of this research. Also for his advice and encouragement provided during the writing process.

Special thanks also to my second supervisor Ann Liggett, for her useful comments and advices during this research.

Special thanks go to special people in my life who have supported me during this long journey with their love, supports, prayers, and encouragements. First, all the thanks to my Parents, Dr. Abdullah Bafail and Mrs. Saffia Samarkandi, without their support none of the work would be possible. A special thanks to my family. Words cannot express how grateful I am to my dear husband, Dr. Fahd Banakhr for all of the sacrifices that he did and for standing beside me during my research to obtain the degree. Also to my beloved kids Rola, Taleen, Ahmed, and Abdullah, who are the source of my happiness and inspiration when I do not feel well.

My thanks also include my brother and sisters, my family-in-law, and all my friends who helped to keep my confidence up and to be there when I needed them.

Finally, the thanks go to King Abdul-Aziz University in Saudi Arabia for their financial support and help to get this PhD degree.

Abstract

Designing for learning is a complex task and considered one of the most fundamental activities of teaching practitioners. A well-balanced teaching system ensures that all aspects of teaching, from the intended learning outcomes, the teaching and learning activities used, and the assessment tasks are all associated and aligned to each other (Biggs, 1996). This guarantees appropriate and therefore effective student engagement. The design and promotion of constructively aligned teaching practices has been supported to some degree by the development of software tools that attempt to support teaching practitioners in the design process and assist them in the development of more informed design decisions. Despite the potential of the existing tools, these tools have several limitations in respect of the support and guidance provided and cannot be adapted according to how the design pattern works in practice. Therefore; there is a real need to incorporate an intelligent metric system that enables intelligent design decisions to be made not only theoretically according to pedagogical theories but also practically based on good design practices according to high levels of satisfaction scores.

To overcome the limitations of existing design tools, this research explores machine learning techniques; in particular artificial neural networks as an innovative approach for building an Educational Intelligence Design Tool *EDIT* that supports teaching practitioners to measure, align, and edit their teaching designs based on good design practices and on the pedagogic theory of constructive alignment. Student satisfaction scores are utilized as indicators of good design practice to identify meaningful alignment ranges for the main components of Tepper's metric (2006). It is suggested that modules designed within those ranges will be well-formed and constructively aligned and potentially yield higher student satisfaction. On this basis, the research had developed a substantial module design database with 519 design patterns spanning 476 modules from the STEM discipline. This is considered the first substantial database compared to the state-of-the-art Learning Design Support Environment (LDSE) (Laurillard, 2011), which includes 122 design patterns available.

In order to have a neural-based framework for EDIT, a neural auto-encoder was incorporated to act as an auto-associative memory that learns on the basis of exposure to sets of 'good' design patterns. 519 generated design patterns were coded as input criteria and introduced to the designed neural network with feed-forward multilayer perceptron architecture using the

hyperbolic tangent function and back-propagation training algorithm for learning the desired task. After successful training (88%), the testing phase was followed by presenting 102 new patterns (associated with low student satisfaction) to the network where higher pattern errors were generated suggesting substantial design changes to input patterns had been generated by the network.

The findings of the research are significant in showing the degree of changes for the test patterns (before) and (after) and evaluating the relationships between the core features of module designs and overall student satisfaction. T-test analysis results show statistically significant differences in the test set (before) and (after) in case of the alignment score between learning outcomes and learning objectives (V1) and the alignment score between learning objectives and teaching activities (V2), whereas no statistically significant difference is seen in the alignment score between learning outcomes and assessment tasks (V3). The network gives an average improvement of 0.9, 1.5, and 0.5 in the alignment scores of V1, V2, and V3, respectively. This resulted in increasing the average of satisfaction scores from 3.3 to 3.8. Accordingly, positive correlation with different degrees between student satisfaction and the alignment scores were suggested as a result of applying the network proposal changes.

EDIT, with its data-orientated and adaptive approach to design, reveals orthodox practices whilst revealing some unexpected incongruity between alignment theory and design practice. For example, as expected, increasing the amount of questioning, interaction and group-based activity effects higher levels of student satisfaction even though misalignment may be present. However, the model is relatively ambivalent towards the alignment of learning outcomes and learning objectives suggesting there is some confusion between practitioners as to how these are related. Also, this confusion appears to persist when defining session learning objectives for different types of teaching, learning and assessment tasks in that the activities themselves appear to be at a higher cognitive level according to Bloom’s Taxonomy than the respective learning objectives (resulting in positive misalignment).

Table of Contents

CHAPTER 4: Data and Methodology

CHAPTER 5: EDIT an Educational Design Intelligence Tool for Supporting Design Decisions

List of Figures

List of Tables

List of Acronyms and Symbols

AI	Artificial Intelligent
ANN	Artificial Neural Network
AANN	Auto-Association Neural Network
AT	Assessment Task
BMU	Best Matching Unit
DAE	Deep Auto-encoder
DBN	Deep Belief Network
EvaSys	Evaluation Surveys Systems
FFMPL	Feed-Forward Multi-Layered Perceptron
LO	Learning Outcome
LObj	Learning Objective
MAE	Mean Absolute Error
MLP	Multi-Layered Perceptron
MSE	Mean Square Error
OBL	Outcome Based Learning
QAA	Quality Assurance Agency
RBMs	Restricted Boltzmann Machines
RMSE	Root Mean Square Error
SCG	Scaled Conjugate Gradient
SD	Standard Deviation
SOM	Self-Organising Map
TLA	Teaching and Learning Activity
VLE	Virtual Learning Environment
S	Overall student satisfaction score
V	Overall module alignment score
V1	LOs and Lobjs alignment score
V2	Lobjs and TLAs alignment score
V3	LOs and ATs alignment score
AVG_V	Average alignment scores

CHAPTER 1: Introduction

1.1 Research Statements

The concept of constructive alignment is considered to be one of the most influential ideas in higher education that promotes deep engagement with the learning material in such a way that learners can demonstrate that they have achieved the stated outcomes (Biggs, 1996). It is therefore surprising that there is not a widespread proliferation of supporting models, frameworks, and toolkits that support practitioners to objectively measure how constructively aligned their educational designs are.

Existing learning design tools in addition to the current state-of-the-art learning design systems, such as the Learning Design Support Environment (LDSE) (Laurillard, 2011) inherently lack objective metric systems that are able to measure the degree to which an educational design is well-formed according to design theories such as constructive alignment. Such a metric system would enable teaching staff to make more informed design decisions such as which profile of activities/assessments to use for a particular learning outcome or need of a cohort. So there is a real need to incorporate a metric system that enables intelligent design decisions to be made based on good design principles. This raises the following research challenges:

- How do we measure the quality of an educational design? i.e. For each learning outcome, how well aligned are the associated learning objectives, teaching and learning activities and the assessment tasks?
- How can such a design metric be used to influence good design decisions?
- How can a system adapt overtime to base its measure of quality not just on theory but also on effective practice? That is, as the system encounters more examples of effective practice it adjusts its internal representations of alignment and its subsequent recommendations for enhancement.
- What correlation can be found between student satisfaction and the alignment metric i.e. between theory and practice? What particular changes in module design elicit high level student satisfaction?

Tepper (2006) proposed a computational model of constructive alignment that combines the principles of constructive alignment with those of generative grammar and linear algebra to compute alignment. Tepper's metric adopts a systematic and structural view of educational design and uses Bloom's Taxonomy as means of quantifying alignment of the four main components of an educational design. Basic classical set theory and linear algebra computations are applied to the generative model to provide numerical measures of alignment for both holistic and individual aspects of the educational design. Although the metric is useful in allowing the quality of educational designs to be measured and thus producing constructively aligned instructional designs, it bases both module structure and the categorisations of the different components on pedagogic theory alone (i.e. outcomes-based approach to module design, components are organised according to Bloom's taxonomy and related according to the principles of constructive alignment). The issue here is that such theoretical principles may not sufficiently reflect effective design practice and therefore it is important for the alignment system to be cognisant of this in order to offer pragmatic and realistic design solutions. In this context, a design decision system is an intelligent agent that recommends alternative design decisions to practitioners to help shape and enhance their teaching (Vialardi, Bravo, Shafti, and Ortigosa, 2009). Developing decision making systems with support of artificial intelligence were documented since the past with the traditional approaches as in (Dufournet, 1987) and more novel approaches as in (Sani, and Aris, 2014). In existing learning design tools, the implementation of such recommending systems or decision making systems is hardly rare or it follows the traditional rule-based approaches based on some educational theories only in offering the recommendation and design decisions. Therefore, these tools are static in nature and cannot allow such a system to adapt to changing practices and be tolerant of variations which may actually be successful in practice. Thus this research investigates a novel approach to incorporate theory and practice together to underpin an intelligent tool that can base its measure of quality not just on theory but also on effective practice. So it adjusts its internal representations of alignment and its subsequent recommendations for enhancement.

1.2 Aims, Objectives, and Methodologies

1.2.1 Aims

The overall aim of this research is to provide more intelligent mechanisms that can aid in the formation of more effective learning designs by developing an intelligent alignment tool that is underpinned with good design principles according to constructive alignment theory and good design practices according to high student satisfaction scores.

1.2.2 Objectives

1. Review and analyse existing learning design systems and identify their current limitations;
2. Evaluate and implement the constructive alignment metric developed by Tepper (2006) to compute alignment between the educational components of a module design – this will form the metric engine;
3. Establish a clear design methodology for integrating the metric engine with the current state-of-the-art learning design tools such as the Learning Design Support Environment (LDSE);
4. Design and implement a prototype of the metric engine to read, analyse, and modify design patterns produced by LDSE;
5. Extend the alignment metric to incorporate good design practices based on high student satisfaction scores;
6. Generate appropriate data sets of realistic learning design patterns from good practices associated with high levels of student satisfaction;
7. Identify acceptable and allowable alignment threshold values based on effective practice (good module designs with high satisfaction scores);
8. Incorporate an adaptive engine into the alignment metric;
9. Investigate the use of auto-encoder neural networks trained with back-propagation (and variants thereof) to learn features of good design patterns to form a knowledge-based system that can be used for pattern association;
10. Develop and produce an education design intelligent tool *EDIT* that can measure alignment between core elements of educational design and recommend changes to enhance designs;
11. Investigate such relationships between theory and practice.

1.2.3 Methodologies

This research investigates the synergy of machine learning techniques and constructivist learning and teaching theory to formulate an intelligent metric tool for constructive alignment. Throughout this research several methods are used to collect appropriate data sets for the machine learning task. These methods consist of analysis of the literature and the use of students' conceptions to identify good and effective design practices through the use of their satisfaction scores. Module design desk-based research study, checklist, and in-depth observation are used to collect design pattern data according to differing levels of student satisfaction for the purpose to learn from learning designs of differing quality. Subsequently, the methodology for utilizing machine learning techniques involved the training of various feedforward multi-layer perceptron networks as auto-encoders. Neural auto-encoders also known as auto-associative networks (Bengio, 2009) and they attempt to reconstruct the input data at the output layer. In other words, these networks are trained to remember and associate a number of input patterns forming a perfect memory of training patterns. Thus, when a noisy pattern is presented, the networks associate it with the nearest one in their trained memory. This is useful for restoring or correcting noisy patterns making them well suited for the research problem. For this, different types of auto-encoder architectures were used and discussed in Chapter 5.

1.3 Research Contributions

The major contribution of this research is that it is the first in investigating the usefulness of such an alignment metric that can measure numerically how well aligned the components of an educational design are. This is achieved by producing a meaningful alignment system where acceptable ranges are based on good teaching practices, which are based on high level of student satisfaction, rather than theory alone. The research also investigated a novel approach that utilizes a machine learning approach in the form of artificial neural networks and in particular auto-encoder networks that extended the alignment system to be informed based on the theoretical framework of constructive alignment and also to be informed by good practice examples. This is highly innovative and to date, no other learning design system, including the LDSE, is able to do this. The results of this research identified core design principles inferred from the neural network to form educational designs that can attract higher level of student satisfaction and thus high student engagement. For example, good practice was found to be around the use of high-level activities such as the use of collaborative-based learning and active learning. In addition, positive correlation was found between module designs and student satisfaction when one or more of the higher level activities are associated with learning objectives of different Bloom's levels as it seems better suited in practice and student satisfaction.

In addition to the above contributions, the research also developed a substantial module design database with more than 500 design patterns for the science and technology sector, which have been generated from real and effective module designs that have been evaluated with high students' satisfaction rates. This work is the first to develop such a large design pattern database in which these design patterns are provided in a structured way – so that relations between design components are easily understood and can thus be utilized by other researchers to evaluate their educational design tools and patterns.

1.4 Research Structure

Chapter 2 of this thesis presents a literature review of the existing learning design systems along with their theoretical foundations, functionalities and limitations. The conceptual model of the alignment metric is also presented in this chapter. Chapter 3 identifies the functional requirements of the metric engine and details the process for integrating it with the LDSE to form a software system that is able to compute and display alignment. In chapter 4, the data and research methodology is given. Chapter 5 investigates the different types of neural network methods applied to learn the task of auto-associative. Chapter 6 reports on the outcome of the best network including results generated from the system and subsequent analysis and evaluation of the model's performance. The chapter also discusses the network's underlying 'design preferences' it has discovered from the data. Chapter 7 concludes the presented work and summarises the key contributions made by this research, highlights the limitations of the work, and proposes some suggestions for future research.

CHAPTER 2: Literature Review

2.1 Introduction

Designing for learning is a complex task and considered one of the most fundamental activities of a teaching practitioner (Cameron, 2008). Learning materials need to be designed effectively in order to achieve a well- balanced learning and teaching system. A well-balanced teaching system ensures that all aspects of teaching, from the intended learning outcomes, the teaching and learning activities used, and the assessment tasks, are all associated and aligned to each other (Biggs, 1996). This guarantees appropriate and therefore effective student engagement. Well-balanced (or *constructively aligned*) teaching practices have been proven to foster deep student learning (Marton and Saljo,1976) and thus impact highly on student satisfaction as clearly documented by researchers such as (Arbaugh, 2014) and (Rienties, Toetenel, and Bryan, 2015). The design and promotion of constructively aligned teaching practices has been supported to some degree by the development of software tools that attempt to support the teaching practitioner in the design process and assist them in the development of conscious and purposeful teaching. With the creation of different toolkits in this domain, tools differ significantly in terms of how they are structured and the types of pedagogical patterns provided to aid the design process. Some tools are based on particular pedagogic models or philosophies; others provide structured patterns to guide the teaching practitioners through particular aspects of the design process and to support them in making informative design decisions (Conole, 2013). This chapter will review first the concept of learning design with respect to learning design patterns, pedagogical patterns and their theoretical foundations and how they can assist in the development of effective teaching and learning design. Following that, a selection of some existing learning design tools will be reviewed and analysed by describing what they are, reviewing their functionalities, the guidance and support they provide during the design process, and their limitations. The chapter will also review the concept of "constructive alignment" and using it in outcome-based teaching and learning and how it is possible to objectively measure the degree to which a module design is well-balanced and subsequently achieves constructive alignment.

2.2 What is a 'Learning Design' and Why Learning Design Tools? Clarifying the Concept

The term *'learning design'*, or as some prefer to use other terms, such as 'educational design', 'instructional design' or 'curriculum/course design', all the terms tend to focus on the importance of 'design' and have a variety of definitions and interpretations within the literature. James Dalziel (2009) defines learning design as *"A process that describes how educators make decisions about creating effective teaching and learning experiences".* Conole (2007) defines the concept as *"a methodology that has emerged in recent years as a semi-formal process for supporting the curriculum design process";* Beetham and Sharpe (2007) refer to learning design as the range of actions or elements associated with creating learning activities that students undertake to achieve a set of intended outcomes. Mor and Craft (2012) simply define learning design as a process of planning new practices by planning activities, resources, and tools that can help at achieving particular educational aims in a given situation. All these definitions revolve around the process of planning, structuring and sequencing learning activities to deliver effective learning experience. Conole (2008) mentioned that one of the approaches to learning design is to adopt a more general interpretation of learning design – one that focuses on pedagogy and the activity of the student. This approach advocates a process of 'design for learning' by which one arrives at a plan, structure or design for a learning situation, where support is realised through tools that support the process (e.g. software applications, websites) and resources that represent the design (e.g. designs of specific cases, templates).

2.3 Representing the Learning Design

Learning design can take place at a number of different levels and can be represented in different forms to offer teaching practitioners different insights into their designs. The type of insight offered may include modelling the kind of learning experience that their students might have; sequencing the teaching and learning activities visually in user-friendly interfaces; or representing the learning activities in some notational format so that it can serve as a model or template to guide the creation of the learning design. One of the most popular approaches to representing learning designs is the application of design patterns and pattern languages as derived by Alexander (1977). The concept of design patterns and in particular pedagogical patterns has strong similarity/association with learning design tools in assisting the development of effective teaching and learning designs because they capture successful solutions by providing an overall structure in a basic format that describe the core features of context, problem, and solution. The design patterns approach identifies the core components of the learning design in more generic descriptors as described by Alexander, a design pattern is *"a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice"* (Alexander et al. 1977). Pedagogical patterns on the other hand are more high-level patterns which seek to find the most effective approaches to teaching by capturing the core design property of a teaching-learning activity as illustrated in Bergin's description, a pedagogical pattern is *"the intent is to capture the essence of the practice in a compact form that can be easily communicated to those who need the knowledge. Presenting this information in a coherent and accessible form can mean the difference between every new instructor needing to relearn what is known by senior faculty and easy transference of knowledge of teaching within the community"* (Bergin, 2012).

Laurillard and Ljubojevic (2011) define the pedagogical pattern as *"a teaching-learning activity sequence that is designed to lead to a specific learning outcome*". They differentiate pedagogical patterns from the generic design patterns in that the fundamental idea of a pedagogical pattern is designed to capture, test, and share best practice of teaching. The structure of a pattern represents teaching practice in terms of pedagogical properties associated with the teaching and learning activities. It combines the general design criteria, the pedagogical properties of the teaching and learning activities such as group size, duration, etc. and the capabilities of tools, resources, and technologies being used. Thus, the structure of the pedagogical pattern is superior to that of a design pattern and enables the learning design to be subjected to computational analysis (Tepper 2006, Laurillard, 2012).

The main elements of a typical pedagogical design pattern are as follows:

- module/session name;
- start and end dates;
- elapsed time;
- learning time;
- number of students;
- topics;
- aims;
- learning outcomes;

- assessment task;
- teaching and learning activities.

Several tools have been developed to facilitate the learning design process itself and to help the designer/practitioner in planning learning outcomes, activities, assessments and other aspects of the learning. The JISC Design Studio (2013) stated that the main purpose of these tools is to allow teaching practitioners to plan learning from small-scale activities up to whole lessons and modules and to support both design and delivery. All learning design authoring tools generally have the common goal of facilitating the process of selecting, structuring, sharing, and reflecting on the learning design; however, different tools support different learning approaches, use various representations, and operate at different levels of granularity from simply capturing the essence of a design to aiding in its semi-automated enactment with students (Conole, 2007, 2010; Dalziel, 2009; Cameron, 2011; and Laurillard, 2012). The tools attempt to support teaching practitioners in designing their activities by providing step-by-step guidance and support at different levels (from a simple learning activity, session, module to a whole course).

2.4 Overview of Some Existing Learning Design Tools

The development of the various learning design tools is to support the learning design process itself and to assist teaching practitioners in planning learning outcomes, activities, assessments and other aspects of the learning and teaching process. As mentioned previously, the wide spread of different learning design authoring tools generally have the common goal of facilitating the process of selecting, structuring, sharing, and reflecting the learning design; however, they adopt different approaches to learning design and differ significantly in terms of how they are structured and the way their pedagogical patterns aid the design process. Some tools are based on particular pedagogic models or philosophies such as the LDSE; others provide structured patterns to guide the teaching practitioners through particular aspects of the design process and to support them in making informative design decisions (Conole, 2013). For this purpose, the following sub-sections will review existing notable learning design and pedagogical planner tools that have been developed to support the design process. It will begin by describing what they are, reviewing their functionalities, their learning design representation, the guidance and support offered during the design process and their limitations. Table [1] at the end of this section summarises the differences among these tools. The section will focus on a number of specific pedagogical tools in details; namely the Learning Activity Management System (LAMS)[1], Phoebe[2], the London Pedagogical Planner (LPP)[3], and the Learning Design Support Environment (LDSE)[4]. These models were selected for review on the basis that they generally reflect the state-of-the-art in educational design tools and are the most highly developed systems of their kind. It is, however; worthy to note at this stage, that there are other models of interest that were briefly considered, namely: LearningMapR[5], Learning Designs[6], RELOAD[7], and CompendiumLD[8]. These models were all developed to help teaching practitioners to create, represent, visualise, deliver, and exchange their learning design. However, they are precluded from the detailed review here for a number of important reasons. Firstly, the most important issue is that these tools are focused on supporting the development of session plans without clear guidance based on supporting pedagogic principles. There are no recommendations for enhancement suggesting the underlying representation of what good practice is. Secondly, the tools typically suffer from poor usability and overly complex user interfaces making it difficult to understand the design process to create meaningful and usable design pattern. Finally, the uptake, impact, and evaluation of these tools are insignificant and in some cases, non-existent.

2.4.1 Learning Activity Management System (LAMS)

LAMS is one of the earliest examples of a learning design tool for supporting the learning design. It is developed by James Dalziel and his team based at Macquarie University, Australia as part of the Macquarie E-learning Centre of Excellence (MELCOE, 2009). The tool is described as: "*A software tool for designing, managing and delivering online collaborative learning activities. It provides an easy to use visual authoring environment to create sequences of learning activities*" (Dalziel, 2003). The tool facilitates the micro-level planning and automation of the learning activities by providing a palette of activities which teaching practitioners can use to drag and drop activities from the palette to the main design area. These activities are then connected together to create a learning activity sequence as

[1] LAMS http://www.lamsinternational.com/
[2] PHOEBE http://www.jisc.org.uk/publications/reports/2008/phoebefinalreport.aspx
[3] LPP http://www.jisc.ac.uk/publications/reports/2008/llpfinal.aspx
[4] LDSE https://sites.google.com/a/lkl.ac.uk/ldse/
[5] LearningMapR http://www-jime.open.ac.uk/article/2005-17/294
[6] Learning Designs http://www.learningdesigns.uow.edu.au/
[7] RELOAD http://www.reload.ac.uk/
[8] CompendiumLD http://compendiumld.open.ac.uk/

shown in Figure [2.1]. The tool also enables the user to create lesson plans in a standardized template that can be easily modified and reused. It provides templates based on best practice processes that include advice on using and repurposing these templates for different learning contexts. The advantages of the tool are: first it provides a high degree of flexibility as it is graphically-based with an easy to use interface for both technical and non-technical users helping them in creating and delivering learning activities through integrated monitoring panels (Britain, 2004; Cameron, 2011 and Dalziel, 2009). Second it provides a fully functionally runtime environment that allow real-time monitoring of the performance of learners (Britain, 2004; Conole, 2007). However, the tool is limited to session level only and sequencing activities within the session. It does not operate at module or course level and neither does it consider other important components of an educational design such as learning outcomes (Cameron, 2011). Clearly, there is no functionality within LAMS that links the activities to the learning outcomes or objectives of an educational design making it of limited use to practitioners designing whole modules and courses that require holistic design decisions.

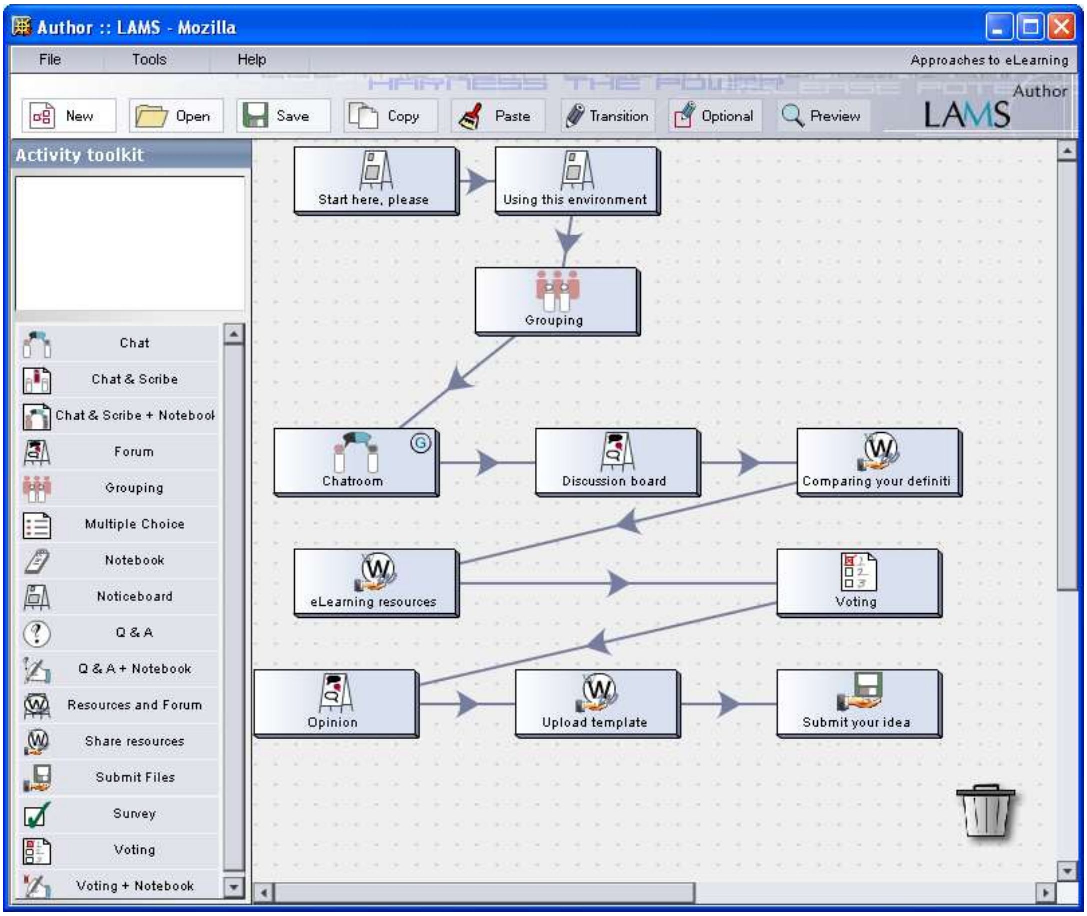


Figure [2.1]: LAMS's authoring environment representing the sequences of learning activities in workflow-style. (Source: http://www.lamsinternational.com/)

2.4.2 Phoebe

Phoebe is a simple decision support tool, developed by Marion Manton, Liz Masterman, and David Balch from the Technology-Assisted Lifelong Learning unit (TALL) at Oxford University and Oxford University Computing Services (OUCS). The tool attempts to provide a comprehensive online resource of tips and hints to support decision-making. The tool provides the following key functionality for the user: create or modify designs, view shared designs, browse Phoebe's teaching and technology guidance, and manage a design template. Figure [2.2] provides a simple schematic diagram of the functional characteristics of Phoebe and how it is used. The tool acts more as a simple authoring environment which allows the user to create learning designs from pre-defined templates. As the user works through a design they are supported by access to context-specific help, wider guidance and resources. The tool includes an extensive wiki of support and guidance on learning design and provides information about the different pedagogic approaches and different digital tools to support different learning activities. The Phoebe tool brings together the key components of a learning design (or lesson plan), prompts teachers' thinking, allows them to record ideas and requirements, and makes it easy to cross-reference components as they design the activities that make up a learning experience. It offers both flexible and guided paths through the planning process, and provides access to a wide range of models, case studies and examples of innovative learning designs.

The key strength of Phoebe is the considerable amount of information that is available to guide the user through completing the various steps of the design. The guidance includes information on: contextual information associated with the design, learning outcomes, assessment, the characteristics of the learners, possible learning activity sequences, contingencies to take into account, reflection and other web links. On the other hand; the tool suffers from some drawbacks in terms of the non-intuitive user interface, and sequential navigational route for the design process Conole (2013). Conole also points out that Phoebe is more text-based that would be best suited for teaching practitioners who adopt a systematic approach to their design practice rather than other approaches that adopt the feature of visual representations of the learning design like some other tools discussed elsewhere in this chapter. Although that the tool is holding different templates in different formats, it does not provide a measure of alignment or guidance to what good design pattern is.

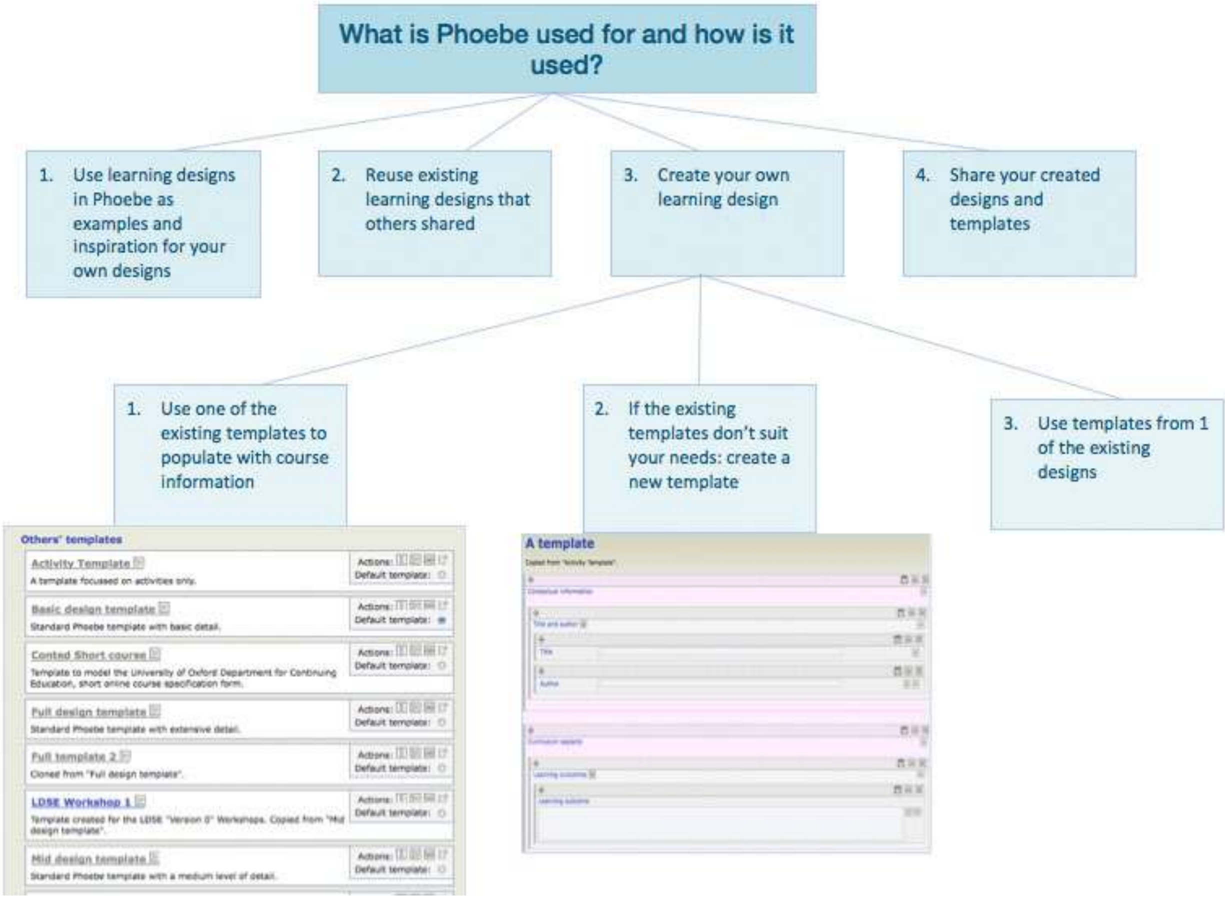


Figure [2.2]: Phoebe's schematic diagram of what it is and how it is used.

(Source: http://www.jisc.org.uk/publications/reports/2008/phoebefinalreport.aspx)

2.4.3 London Pedagogical Planner (LPP)

LPP is a structured modelling tool that supports lecturers in developing, analysing and sharing learning designs. It was developed by Diana Laurillard and her team at London Knowledge Lab in the Institute of Education. The tool enables teaching practitioners to map different teaching methods to five types of cognitive activities, namely attention, inquiry, discussion, practice, and production. Users can link between aims, outcomes, teaching methods, topics, assessment and then map topics and associated learning outcomes across different blocks of study. The tool supports planning at both the module and session level. It enables teaching practitioners to first provide general information about their learning design and then it supports them by ensuring that all topics and elements are mapped as seen in Figure [2.3]. After this point, users can then enter the amount of time needed for each of the different types of activities and then map topics to a calendar and allocate the number of hours across the types of activities and the topics across the different types of activities. The tool then calculates a statistical description of the learning design against the different types of activity defined (Laurillard, 2007). The descriptions produced are numerical in terms of the likely amount of time is allocated to the selected activity methods with the different cognitive activities they elicit and graphical representation of the learning design (Laurillard, 2007). The tool aims to make the pedagogical design explicit as an output from the process, capturing it for testing, redesign, reuse and adaptation by others. One of the drawbacks of this planner, as mentioned by Conole (2013), is that the approach of the tool is likely to lead teaching practitioners to replicate existing practice rather than change their practice. The reason for this is that the statistical information given is merely descriptive and does not provide a critical account or a means of adapting the learning design.

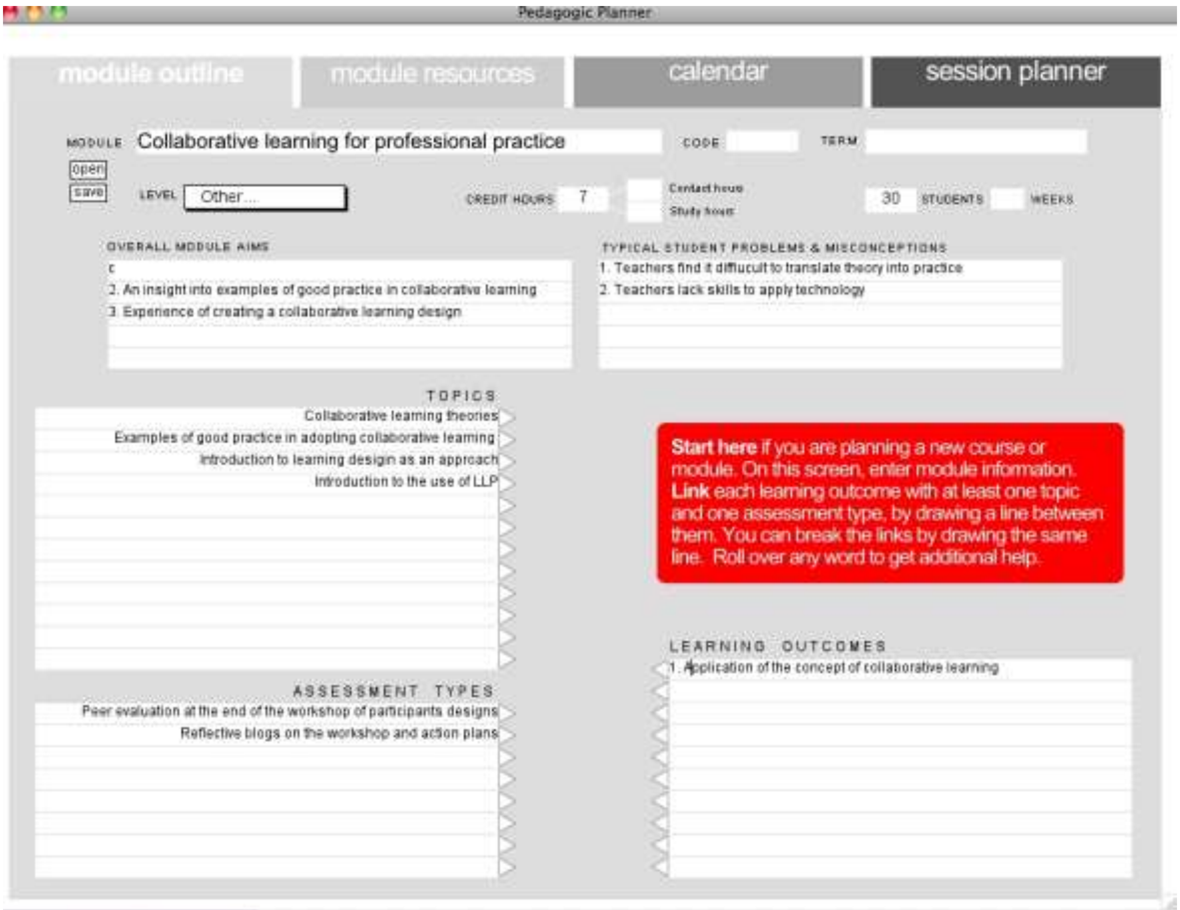


Figure [2.3]: LPP’s general module information.

(Source: http://www.jisc.ac.uk/publications/reports/2008/llpfinal.aspx)

2.4.4 Learning Design Support Environment (LDSE)

LDSE is the successor to the two previously mentioned pedagogy planner tools the LPP and Phoebe. It was developed by Laurillard and a team of researchers at the London Knowledge Lab involving the Institute of Education Birkbeck College, University of Oxford, London Metropolitan University, London School of Economics and Political Science, the Royal Veterinary College and the Association for Learning Technology (ALT). The LDSE project's primary aim is to design a software tool that helps teaching practitioners to design effective technology-enhanced learning (TEL) programmes. It also supports teaching practitioners to create, modify, share, reuse learning designs, and build on the work of others at the level of module and session design. The tool is based on the following theoretical principles: social constructivism, collaboration, constructionist learning and knowledge building (Laurillard and Masterman, 2010) and the pedagogic principles underpinning the operation of the LDSE have been based on the Conversational Framework developed by Laurillard (2002). The tool incorporates a sophisticated knowledge base in the form of an ontological model[9] that holds the core learning design concepts and their relationships which are used to categorise each learning design imported, adapted, or created within the LDSE environment. This enables the LDSE to enhance the user's experience by providing guidance and recommending alternative suggestions (Laurillard, 2011). This is the key advantage over the other planning tools reviewed. The advice and guidance in supporting teachers is manifested in thinking about using technology enhanced learning activities and reflecting the potential impact of this on their learning designs thus offering a fertile space for reflection. For each of the activities the tool provides alternative ideas and activities, enhancing the balance between acquisition, inquiry, practice, production and discussion. Figures [2.4 & 2.5] illustrate the advice, suggestions, and feedback analysis made by the LDSE.

The LDSE uses a pedagogical pattern and defines a set of core pedagogical properties associated with the teaching and learning activities, so that the teaching and learning activities in a pedagogical pattern can be mapped to the different types of learning in the Conservational Framework (Laurillard, 2012). The structure of the LDSE pedagogical pattern

[9] Ontologies: "*A formal explicit description of concepts in a specific domain, which provide a machine-readable and shared view on conceptualization of domains of interest for a group of systems and human users*" (Charlton & Magoulas, 2010). In LDSE, the ontological model is a knowledge base, which is underpinned by the Conversational Framework (CF, see Laurillard, 2009), that define and hold the core learning design concepts and their relationships. It is used to inference the set of concepts to help the users to complete their learning design based on their current context.

includes the general design descriptors as illustrated below (aims, learning outcomes, assessment tasks), and the pedagogical properties of each teaching and learning activity includes the size of the group, the duration of the activity, and the type of learning it supports. The pattern is then interpreted in terms of the TLAs and the pedagogical properties that include representing the learning design in terms of the proportion of time spent learning through different cognitive activities. These proportions are calculated from the properties of the TLAs designed on the timeline. This approach is based on the conceptual classification of types of teaching and learning activities into five categories: Acquisition, Discussion, Inquiry, Practice and Production (Laurillard, 2010, 2011).

The design elements in LDSE pedagogical pattern are:

- module/session name;
- start and end dates;
- elapsed time;
- learning time;
- number of students;
- topics;
- aims;
- learning outcomes;
- assessment task;
- Teaching and learning activities

Teaching practitioners input all the elements in the pattern template except for the teaching and learning activities which can be dragged and dropped into a 'timeline view' of a session enabling them to see the set of scheduled activities. The teaching and learning activities can be unpacked further to some logistical and pedagogical properties that can help in analyzing the design in terms of the type of learning that occurs in each teaching activity.

The Conversational Framework

The Conversational Framework proposed by Diana Laurillard (2002), is a comprehensive model that focuses on social learning theories (Piaget, 1970; Kolb and Fry, 1975; Papert and Harel, 1991; Roschelle and Teasley, 1995; Shaw and Shaw, 1999) and technologies. The framework provides a representation of what it takes to learn based on the main theories of teaching and learning, such as social constructivism, collaboration, constructionist learning and knowledge building (Laurillard and Masterman, 2010) to break learning down into the essential components needed to create a meaningful learning environment. These theories of learning underpin the use of the LDSE and each one is briefly described below:

- Social constructivism: 'the members of the community serve as active agents in the construction of outcomes and activities that produce a developmental cycle' (Shaw and Shaw, 1999);
- Collaboration: 'a coordinated synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem' (Roschelle and Teasley, 1995);
- Constructionist learning: 'building knowledge structures in a context where the learner is consciously engaged in constructing a public entity' (Papert and Harel, 1991);
- Knowledge building: 'the capacity to create new knowledge and ideas… collaborative problem-solving… needs optimal environments for knowledge-building' (Scardamalia, 2010).

The aim of the Conversational Framework is to support, evaluate, and represent learning designs in such a way that teachers and learning designers can use it to evaluate their learning designs and analyse the overall learning experience and the use of new technologies in learning in terms of the five key types of cognitive activity (i.e. acquisition, inquiry, discussion, practice, and production) (Laurillard, 2002). The conceptualization of the LDSE is expressed in terms of the Conversational Framework, which provides conceptual depth and perspective round a number of the pedagogical theories underpinned the LDSE. The Framework links a unit of learning into the broader ideas of a constructivist perspective which supports the ontological design of the system. This ontological model holds the pedagogic principles and core learning design concepts and their relationships which are introduced through the nature process of constructing a sequence of teaching and learning activities Laurillard (2011).

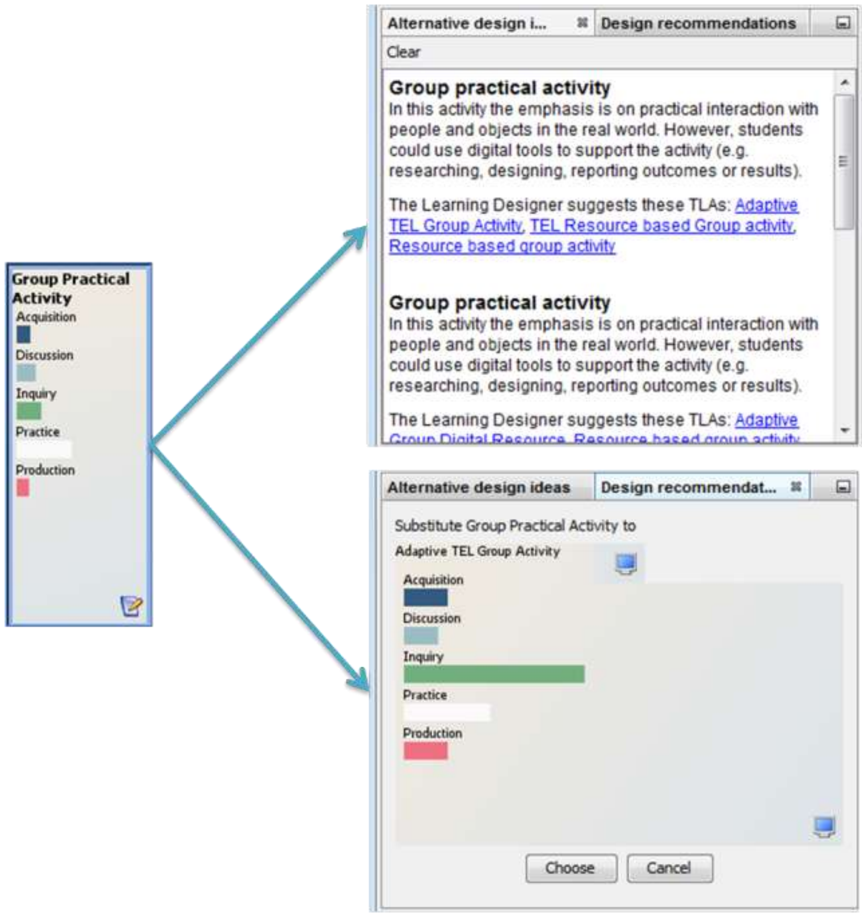


Figure [2.4]: An example of the LDSE suggesting alternative TLAs, to enhance the balance between acquisition, inquiry, practice, production and discussion.

(Source: https://sites.google.com/a/lkl.ac.uk/ldse/)

Figure [2.5]: The analytical representation of the learning design in LDSE in the form of automatically-generated summary statistics. (Source: https://sites.google.com/a/lkl.ac.uk/ldse/)

Summary of the core LDSE Features are:

- Time Modeller to analyse the effects on learning quality, in terms of the impact of cost and time resourcing by a particularly design configuration (Laurillard and Ljubojevic, 2011);

- Community Knowledge-based built into LDSE system to support the development and maintenance of an ontological model of a learning design (Charlton and Magoulsa 2010; Laurillard and Ljubojevic, 2011);

- Inference algorithm to search ontological concepts, classes, and properties within its knowledge base to make inferences and provide guidance and recommendations for optimal ways of combining conventional teaching-learning methods with a variety of technology enhanced methods based on the Conversational Framework (Laurillard, Magoulas, and Masterman, 2011).

2.5 Limitations of Current Learning Design Tools

The different learning design tools that have been reviewed (see Table [2.1] for a summary) have indicated their potential to support teaching practitioners by producing largely web-based applications to assist them in creating and structuring their teaching experience. Much of the work has emphasized the cost-effectiveness, efficacy of using technology-enhanced learning, and the significance of using digital technologies in designing teaching and learning (Cameron, 2011; Atkinson, 2011). In addition, Cameron (2011) emphasizes the importance of these tools and indicates that these pedagogical planners do provide valuable support for reflection and exploration, and help scaffold the design of learning activities. In addition, these tools have indicated their potential to advise and guide teaching practitioners pedagogically through the design process. This advice and guidance is manifested in several ways. For example, LAMS has an associated Activity Planner tool that provides a set of good templates which include advice on using, completing, and repurposing these templates for different learning contexts (Conole, 2013). Phoebe was praised for the quality of its guidance as it incorporates a separate wiki-based online resource of tips, guidance, and digital tools to support the decision-making process (Beetham and Sharpe, 2007; San Diego, Laurillard, Boyle, and Llubojevic, 2008; and Conole, 2013). The LPP, in contrast, integrates the support and guidance during the design process and with the actual decision-making process. It takes the user through a series of design decisions using an inspectable and editable model. Subsequently, it then provides numerical and graphical representations of the resulting learning design which is visualized in terms of Laurillard's types of activity (Laurillard, 2002; Laurillard, and Llubojevic, 2010). The LDSE incorporated an ontological model which has been informed both by pedagogic theories and their understanding of lecturers' design practice elicited through interviews with learning design practitioners. The tool is able to interpret, analyse, and calculate the learning experience in terms of only the TLAs and the type of cognitive learning and suggest alternative activities relevant to the chosen activity. This guidance and support is based on drawing inferences from comparisons between the user's design decisions and the developed ontological model formed based on educational theories.

Table [2.1]: Summary of Learning Design Tool

	LAMS	Phoebe	LPP	LDSE
Design level	Session level-planning and automation of the teaching and learning activities (i.e. sequencing activities within a session).	Module and session level.	Module and session level.	Module and session level.
Theoretical framework(s)	Support wide range of pedagogies, including transmission, instructivist, constructivist, PBL, case based.	Not specifically- online support to guide good practice.	Strong focus on categorising the TLAs according to five types of cognitive activities (attention, inquiry, discussion, practice, production).	LDSE based on the Conversational Framework that supports the main learning and pedagogical theories: instructivism, constructionism, social constructivism and collaborative learning.
Design pattern components	TLAs that include a range of individual tasks, small group work and whole class activities based on both content and collaboration.	Aims, outcomes, teaching methods, assessment, learning approach, resources.	Aims, outcomes, teaching methods, assessment, learning approach, resources.	Aims, outcomes, teaching methods, assessments, tools and resources; rich pedagogical properties attached to TLAs; other TLA features such as type, duration and cohort size.

	LAMS	Phoebe	LPP	LDSE
Representation of design	Represent the sequences of learning activities in workflow-style visual authoring interface.	Text-based (text template) with a set of pre-defined templates for user to complete.	Tap/form-based interface allowing user to map different TLAs to five types of cognitive activities. Graphs for visualising how learner time is shared across different TLAs.	Represent the learning design in terms of the TLAs showing pedagogical compositions of learning types (linked to theoretical framework). Graphical statistical summaries of sessions with duration and learning methods used.
Measures design quality?	N	N	N	N
Makes recommendations?	N	N-Wiki-based reference links and resources provided.	N-but users can manually map different design components together.	Y-only recommends alternative TLAs on the basis of the properties of the currently used TLA and suggests ways to combine TLAs and TEL approaches to promote deeper learning.
Adaptive knowledge-base?	N	N	N	N

It can be concluded that the existing tools do provide help and support to teaching practitioners in designing their courses and classes as seen by providing different representations to offer designers different insights into their designs, including modelling the kind of learning experience that their students might have, or sequencing the teaching and learning activities visually in user-friendly interface and providing design decisions with advice on making those decisions. However, the reviewed tools do not offer judgement on how well-balanced or aligned the learning design is with respect to the learning outcomes. The reason for this is that these models inherently lack an objective measure of alignment in order to make such recommendations or judgements. Subsequently, there is a real need for a metric system which is able to measure and compute the degree to which an educational design is well-balanced or well-formed according to set guiding theoretical principles such as those of constructive alignment. Another key limitation is that none provides an indication as to how well the design will work in practice in terms of student satisfaction. The tools are currently based on theoretical principles of good design and the rules that govern the tools behaviour are hard-wired subsequently cannot be adapted according to how the pattern works in practice. Although the LDSE uses AI-based methods and utilises a sophisticated knowledge base, its current limitations are as follows:

- The tool is unable to represent the degree to which an educational design is well-balanced according to the principles of constructive alignment and how well the design works in practice and tailors its recommendations accordingly;
- The tool focuses on recommending alternative teaching and learning activities but is unable to recommend alternative types of learning objectives and assessment tasks relative to the learning outcomes;
- The LDSE knowledge base is based on theoretical principles only and is static such that its symbolic rules are hard-wired (design in) and must be updated manually.

The last point is particularly significant as such a static knowledge base underpinned by theory alone does not allow such a system to adapt to changing practices and be tolerant of variations which may actually be successful in practice. A system that is unable to adapt its knowledge in light of new information will be severely limited in the scope of enhancement decisions it can make and inherently unable to offer pragmatic and realistic design solutions in light of changing practices.

Given the current issues and limitations of the learning design tools reviewed, there is a real need to incorporate a metric system that allows the quality of a learning design to be quantified and on this basis to enable intelligent design decisions to be made based on good design principles and effective practice i.e. as the system encounters more examples of effective practice it adjusts its internal representations of alignment and its subsequent recommendations for enhancement.

2.6 Measuring Constructive Alignment: The Computational Framework to Measure Good Design Practices

Constructive alignment, which is an outcome-based design methodology for optimizing the conditions for quality learning, through its integration of instructional design and constructivist principles, offers a theoretical and practically proven alignment system that can form the basis of a computational system engineered to assist the teacher during curriculum design (Biggs, 2000). Before introducing the computational framework of the alignment metric, this section will address some of the important concepts related to the principle of "constructive alignment". The computational framework is then presented along with its limitations.

2.6.1 Outcome-Based Learning and the Theory of Constructive Alignment

The move to Outcome-Based Learning (OBL) approach has been one of the most important trends in the nature of higher education (HE) in the United Kingdom (UK). For example, Subject Benchmarking, National Framework for HE Qualifications, Personal Development Portfolios and activities of the QAA are all strongly underpinned by the OBL approach and significantly dependent upon learning outcome statements in some form (Jackson, 2000). Higher education institutions and universities have been required to express their courses in terms of course level and module level model learning outcomes statements since the outcomes of the Dearing Report (NCIHE, 1997) were implemented by the government. Fundamentally, the shift to a curriculum framework based on OBL resulted in the need to articulate teaching and learning activities and assessment tasks with respect to the learning outcomes in a way "that will engage students in the activities most likely to lead to quality learning" (Biggs, 1999). The QAA is a national body set up to improve the academic standards and quality of HE in the UK, and has become a major champion for the incorporation of OBL principle in education design and in particular those principles of constructive alignment (Jackson, 2000). The OBL approach is a fundamental feature of the concept of constructive alignment which is considered to be one of the most influential ideas in higher education today (Cohen 1987; Biggs 1996, 2002; Jackson 2000; Biggs and Tang 2011). Jackson (2000) mentioned that the OBL approach presumes that the results of learning can be expressed in a form that permits their achievement to be demonstrated and measured. Many definitions exist to define OBL; one of them is defined by Margery (2003) as *"an approach to education in which decisions about the curriculum are driven by the exit* *learning outcomes that the students should display at the end of the course".* The approach provides a solid foundation for learning, teaching and assessing because it emphasises that the learning outcomes must be clearly defined as should the activities students need to be able to do to demonstrate they have met the outcomes. Therefore, it is essential that teaching practitioners are able to construct and articulate their programme and module designs within an outcomes-based context. Underlying the outcomes approach to defining, designing, promoting and assessing students' learning is a useful theory of learning known as *constructive alignment* (Biggs 1999). The theory connects the abstract idea of a learning outcome to the things teachers actually do to help students learn, and the things that students do to actually learn. The concept of constructive alignment and its principle in the educational design will be discussed in the next section.

2.6.2 The Alignment Principle in Educational Design

Biggs (1996) mentioned that teaching forms a complex system that takes place at class room level, department level, or institutional level. Taking the class room level as a system which comprises the following components teachers, students, curriculum, teaching and assessment tasks where these components must put together working towards an aligned system forming constructively aligned teaching and learning. Constructive alignment is defined as an outcomes-based approach to curriculum design ensuring that components such as aims, topics, learning outcomes, learning objectives, teaching methods, and assessment are all integrated and aligned with each other, forming a cohesive and effective learning design. Biggs based the alignment components of constructive alignment on Cohen's (1987) idea of instructional alignment where the curriculum objectives (outcomes), teaching methods, and assessment tasks need to be aligned leading to "massive improvement" (Biggs, 2002; Cohen, 1987). Biggs illustrated a simple diagram showing an example of an aligned and unaligned course in Figure [2.6] and Figure [2.7].

Three main steps identified in order to construct an aligned system:

1- Define clear learning outcomes by determining what students need to know, do, or understand after the learning has taken place specifying level of understanding. According to Biggs, these learning outcomes can be stated in terms of appropriate verbs where the verb says what the relevant learning activities are that the students need to undertake in order to attain the intended learning outcome. For this a model of understanding, cognition, and quality of learning is needed such as Bloom’s Taxonomy (Bloom, 1959) which helps to map levels of understanding that can be built into the intended learning outcomes and to create the assessment criteria.
2- Select appropriate teaching and learning activities that get students to elicit these outcomes.
3- Set appropriate assessment tasks that addresses the intended outcomes.

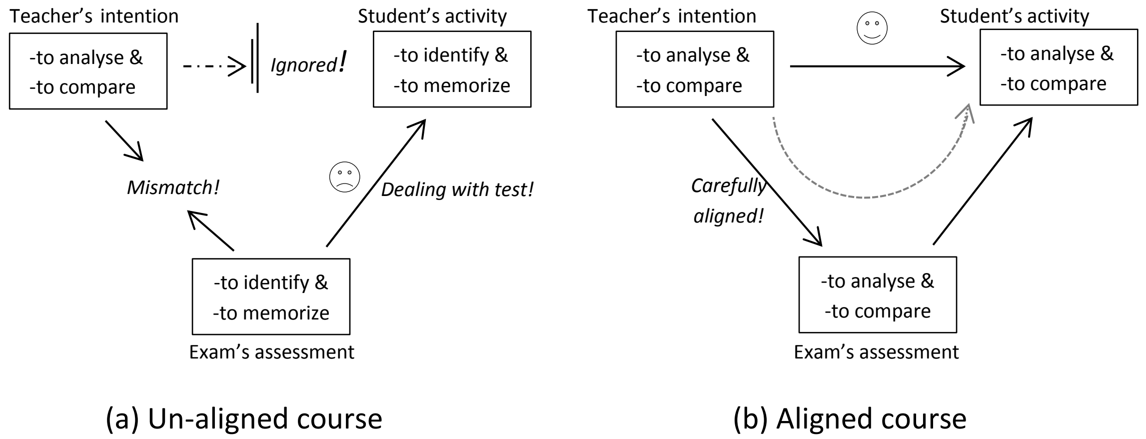


Figure [2.6]: Example of aligned and unaligned course based on Biggs' model (Biggs, 2003).

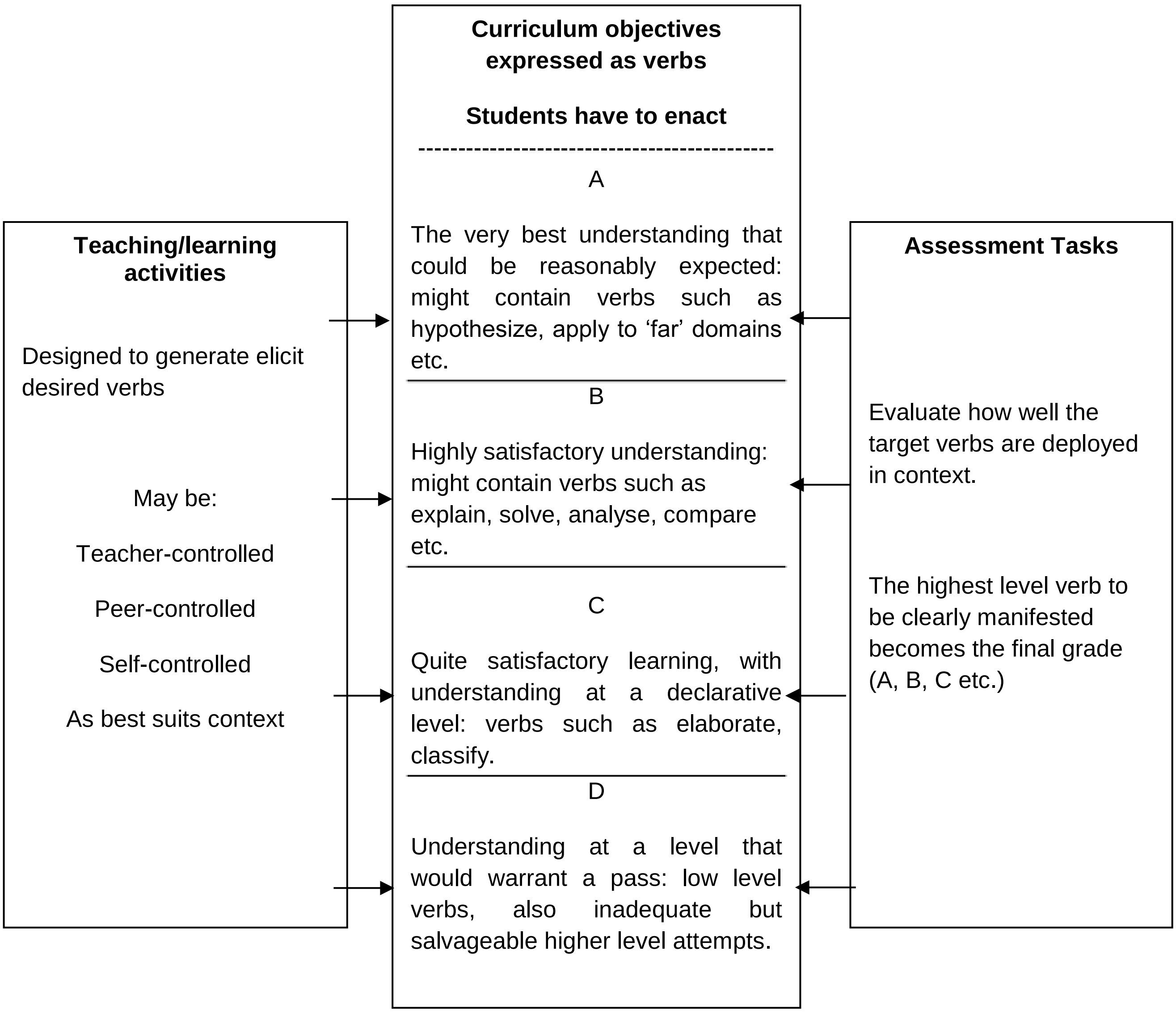


Figure [2.7]: Biggs' model of constructive alignment (Biggs, 2003).

2.6.3 Bloom's Taxonomy

Bloom's Taxonomy (Bloom, 1959) is a broadly accepted classification model for classifying and identifying levels of performance. The model is used as a foundation to support teaching practitioners for classifying cognitive process using action verb statements that can be observed and measured. Also it ensures and assesses the alignment of teaching activities and assessment tasks to learning outcomes. Bloom's Taxonomy, giving levels from high to low, and the corresponding levels of cognitive ability stimulated by a particular action, are shown in Table [2.2]

Table [2.2]: Bloom's Taxonomy and the corresponding levels of cognitive ability stimulated by a particular action (Bloom, 1959).

Higher cognitive abilities

Level	Cognitive Ability Stimulated	Action Elicited	Verbs
6	Evaluation	Ability to make a judgment of the worth of something	Appraise, Assess, Choose, Compare, Critique, Estimate, Evaluate, Judge, Measure, Rate, Revise, Score,
5	Synthesis	Ability to combine separate part into a new whole or propose alternative solutions	Arrange, Assemble, Collect, Combine , Compose, Construct, Create, Design, Devise , Develop, Formulate, Modify, Organize, Plan, Prepare, Produce
4	Analysis	Ability to break down objects or ideas into simpler parts and find evidence to support generalizations.	Analyse, Calculate, Categorize, Compare, Conclude, Contrast, Correlate, Criticize,, Deduce, Debate, Detect, Determine, Develop, Diagram, Differentiate, Distinguish
3	Application	Ability to apply knowledge to actual situations.	Apply, Complete , Demonstrate, Dramatize, Employ, Generalize, Illustrate, Interpret, Operate, Practice, Relate, Schedule, Use, Utilize, Initiate
2	Comprehension	Ability to understand facts and rephrase knowledge	Describe, Determine, Differentiate, Discriminate, Discuss, Explain, Express, Give, Identify, Locate, Report, Review, Restate, Recognize, Select, Tell, Translate
1	Knowledge	Ability to remember previously learned information	Arrange, Define, Identify, List, memorise, Name, Recall, Recognize, Record, Relate, Repeat

Lower cognitive abilities

2.7 The Computational Framework for Measuring Constructive Alignment (Tepper, 2006)

The computational model of constructive alignment developed by Tepper (2006) combines the principles of constructive alignment with those of generative grammar and linear algebra to compute numerically the alignment and measure how well aligned the educational components are when put together. Tepper's metric adopts a systematic and structural view of educational design and uses Bloom's Taxonomy as means of quantifying alignment of the four main components of an educational design. Basic classical set theory and linear algebra computations are applied to the generative model to provide numerical measures of alignment for both holistic and individual aspects of the educational design.

The aim of Tepper's alignment model is to act as a framework to assist teaching practitioners to consistently and systematically produce constructively aligned curricula. The metric developed was inspired by Bloom's Taxonomy and its variants (Bloom, 1956; Anderson el al., 2001) and Bigg's constructive alignment (Biggs, 1996; Biggs, 1999). It uses Bloom's Taxonomy to verify the correspondence between the core elements defined for a specific module or session. It tends to contribute to a more aligned and flexible educational system. Thus for effective learning to take place, there is the need for ensuring constructive alignment of the curriculum, which aims as mentioned before to create a link between the educational design components such that the described intended learning outcomes in the module are supported by the learning objectives, learning activities, and assessed using the suitable assessment tasks. This will help to achieve effective student engagement and well-aligned teaching practice. The main motivation behind the development of the alignment metric model was to address questions to whether such a model or a framework can assist teaching practitioners to quantitatively measure the level to which their module design is constructively aligned; and to develop and design a constructively aligned curriculum that is fair to all students and enforces inclusivity.

The core components of the alignment metric are considered to be the same core components of any educational design which are: the learning outcomes (LOs), learning objectives (LObjs), teaching and learning activities (TLAs) and assessment tasks (ATs). Each component is briefly described and Figure [2.8] below shows how these components inter-relate in a systemic way.

Learning outcomes are statements that describe what the students 'should' have learnt having completed the teaching and learning activities;

Learning objectives are teacher-orientated and/or student-orientated statements that specify what activities the students need to perform to achieve the associated learning outcomes. They determine the teaching and learning activities (TLAs) used. When defining learning objectives it is essential to consider the existing knowledge and experience of the typical student entering the module.

Teaching and learning activities (TLAs) refer to teaching methods and techniques that are chosen to get the students to do what the learning objectives nominate.

Assessment tasks (AT) Formative assessment tasks refer to those student activities that provide an indication (to student and teacher) as to how well the student is developing and attract no formal marks. Summative assessment tasks refer to those student activities that reveal how well they have met the intended learning outcomes and are used to make official judgments about student performance and attract formal marks.

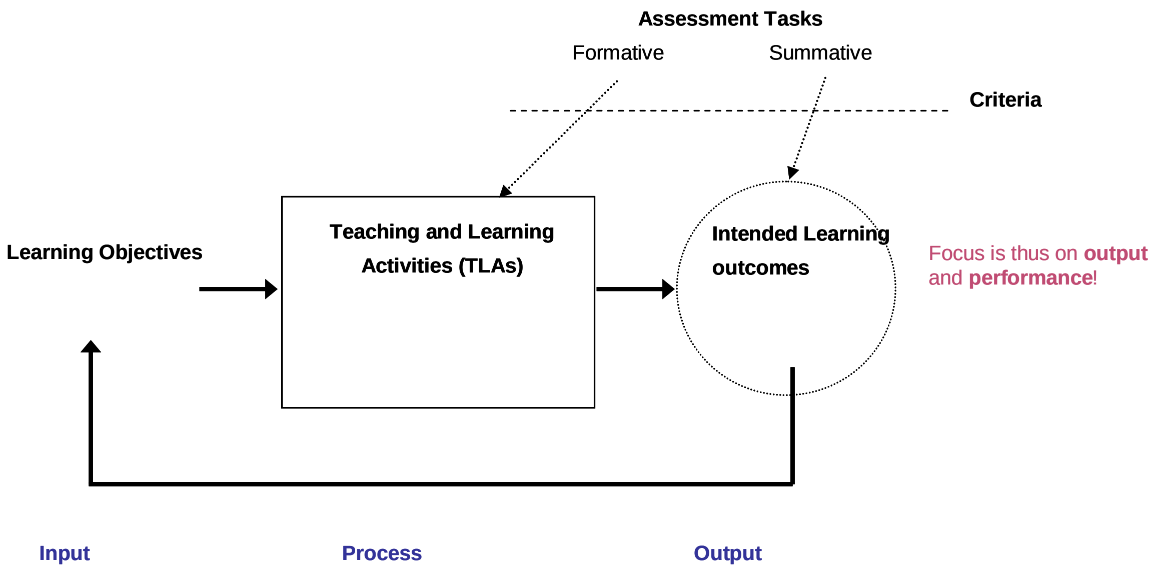


Figure [2.8]: Components of a module design and how these components inter-relate in a systemic way (Tepper, 2006).

The computational model combines the work of Biggs (2002) on Constructive Alignment by adopting a systematic and structural view of educational design, and using Bloom's Taxonomy as a quantifiable measure of the four main components of the educational framework (*learning outcomes (LOs), learning objectives(LObjs), teaching and learning activates (TLAs), and assessment tasks (ATs)).* It applies Set Theory to represent the relations between components and linear algebra to compute the alignment. It provides a numerical measure of alignment for both holistic and individual aspects of an educational design. Figure [2.9]: below indicates the framework relation between components of the model generating three distinctive sorts of tree structure:

a- Learning Outcome tree (Lo)
b- Learning Objective tree (Lb)
c- Assessment Task tree (AT)

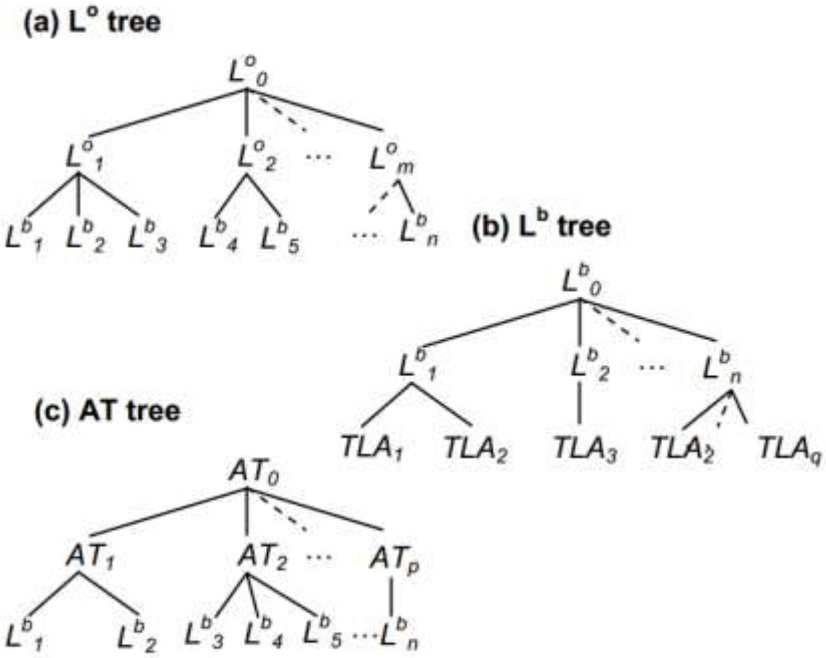


Figure [2.9]: Different types of tree structure a) Lo tree showing relationships between outcomes and objectives; b) Lb tree showing relationships between objectives and TLAs and c) AT tree showing relationships between ATs and objectives (Tepper, 2006).

Considering holistically for an entire module or programme, the generative system would generate a structural perspective of the tree structure allowing the relationships between all the system components for a single learning outcome to be represented and manipulated in vectorial form as in Figure [2.10].

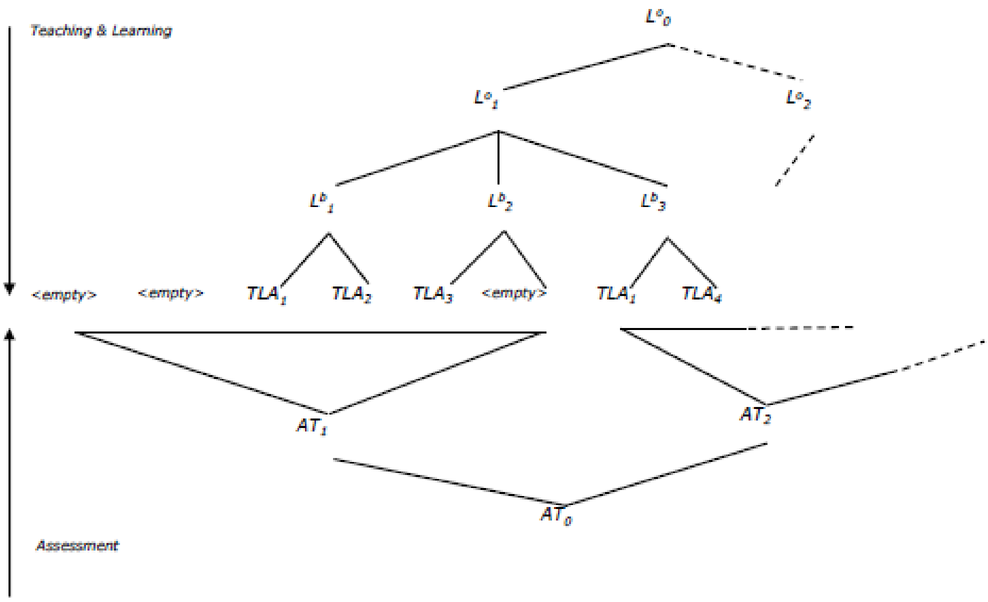


Figure [2.10]: Structural perspective of balanced tree structure showing relationships between system components for a single LO (Tepper, 2006).

For the purposes of aligning elements between the four sets outlined, the model adapted Biggs' method for utilizing verb-matching plans to figure out the level of cognitive capability managed by an assessment task and gives a broad list of suitable assessment tasks for the distinctive sorts and levels of studying needed by a learning objective (Biggs, 1999). The verb-matching schema was used to assign each textual definition of an outcome, objective, TLA and AT to a suitable level in Bloom's taxonomy. The level is acquired by matching the fundamental verb in the result or objective with the comparing section in Bloom's taxonomy that holds a matching or synonymous verb. Categorizing the system components and identifying the relation between them enable the model to perform the mathematical computational required to compute and measure the alignment.

The model computes the alignment across the trees structures outlined above in Figure [2.9] to yield alignment between individual components for example, alignment between Learning outcomes and learning objectives (Lo tree), Learning objectives and TLAs (Lb tree), ATs and Learning objectives (AT tree). It calculates the degree for achieving the highest level (ideal alignment) and thus an educational design is to be considered constructively aligned when all components have reached their equilibrium (Tepper, 2006).

Tepper (2006) in his computational model utilizes two important terms during the alignment computation *'positive misalignment'* and this refers to the situation where one or more of the dominated learning objectives, TLAs, or/and ATs elicit cognitive abilities exceeds the level of the associated learning outcome. Conversely, the term 'negative misalignment' refers to the situation where the educational components elicit lower cognitive abilities than the level of the associated learning outcome. It is negative because although that the learning objectives, TLAs, and ATs may be achievable, the learning outcome itself is still unobtainable.

The strength of the model lies in that the alignment metric is based on the verb where the four components are categorized using Bloom's taxonomy by matching the main verb with the corresponding entry in Bloom's taxonomy. This verb-matching scheme was used as a basis to cluster the different system components according to the cognitive skill elicited. This utilizes such principles to enforce the selection restrictions based on learning elicited. The strength of the model also lies in facilitating and supporting teaching practitioners to adapt their practice to better align their modules by making them aware of alignments and misalignments within their educational designs. Thus it has been established that the level of constructive alignment can be measured using vectorial representations and computations to provide numerical measures of alignment between individual system components and for an entire module or session.

2.7.1 Limitation of the Alignment Model

As discussed in the previous section, the alignment metric provides a quantitative measure of alignment between individual system components and of full constructive alignment for an entire module based on theory of constructive alignment and Bloom's taxonomy. This facilitates teaching practitioners to adapt their practice to better align their modules by making them aware of alignments and misalignments within their educational designs. Crucially, the computation of the alignment metrics is dependent upon three important factors: 1) the ability to accurately cluster outcomes, objectives, ATs and TLAs according to the level of cognitive ability they elicit; 2) a priori definitions of acceptable prototypes of perfect or 'desired' alignment values from which to 'benchmark' against; and 3) defining realistic alignment threshold values, which are currently theoretically based i.e. based on theory of constructive alignment and Bloom's taxonomy. However, the metric still lacks true value of alignment and it needs to identify allowable and acceptable values for the alignment thresholds to base its measure on not only theory but also good and effective design practices thus bridging between theory and practice. Extending the alignment metric to be also informed by good practice examples, may help teaching practitioners to better align their module designs theoretically and practically thus to promote deep student learning.

2.8 Eliciting Student's Feedback of Good Learning Design

The approach of eliciting practitioners' conceptions of learning design was utilized by Laurillard and her team (Laurillard et al., 2011). Semi-structured interviews were made with ten 'informant practitioners' in order to elicit their conceptions of good learning design, "desirable" teaching behaviours, and to investigate further the critical characteristics of good pedagogy. This approach is more likely a theory-informed way of measuring the quality of good learning design and does not consider the students' conceptions or their opinions of a good practice, which may be more effective than the conceptions of the practitioners themselves. Many recent studies have revealed that the use of student feedback by measuring the extent of their satisfaction may contribute to enhance the learning experience (Rienties, Li, and Marsh, 2016), as well as it can reveal in the most important aspects of good design that promote deep student learning. Student satisfaction is one of the important indicators of the quality of good teaching practices as indicated by many researchers such as Moore and Kearsley (1996) and Yukselturk and Yildirim (2008). Since the first evaluation conducted at Harvard University in the early 1920s (Remmers, 1926) and other American universities

(Marsh, 1987), the opinions and feedback of students attending university courses have represented the core of the evaluation of the quality of teaching (Solinas, 2012). Various recent studies also have tried to investigate the students' satisfaction as an indicator measure for the quality of teaching system for instance, Arbaugh (2014) and Rienties, Toetenel, and Bryan (2015) have found around 40+ modules where the learning design as well as the teaching support in particular has influenced the satisfaction level of the students. Luigi and Mostafa (2012) collected student feedback after redesigning a power engineering course considering the implementation of constructive alignment, students' level of satisfaction was higher as they found a benefit for their learning within the new course design. In addition, the Student Satisfaction Survey Report (2011) found that the students of the institution under the survey found the 'learning environment' and 'registration' as well as 'recruitment' and 'admission' satisfactory. This gives the opportunity, thus, to make an improvement and set the new teaching objectives. Eliciting students' opinions throughout a course collected from students sheds light on their own perceptions of learning and of the effectiveness of the learning environment created by the instructor, and they are also helpful for ongoing course improvement (Lo, Celia C., 2010). This refers to the fact that the higher the level of satisfaction of students, the higher will be the effectiveness of teaching system showing that students are better satisfied with the teaching methods of the teaching system (Henrad, 2010). For this, a wide range of standardised student evaluation instruments have been developed, such as Course Experience Questionnaire, National Student Surveys, Student's Evaluations of Educational Quality Questionnaire, or Evaluation Surveys Systems (EvaSys). All these methods are designed for the purpose of capturing students' feedback on any issues or areas of best practice regarding different aspects such as the teaching styles, module organisation, module's outcomes clarity, learning resources available, and the overall learning satisfaction (Course Experience Questionnaire (CEQ) Report, 2009) (Student Evaluation Surveys (EvaSys) Nottingham Trent University, 2014). Studies revealed the potential use of these evaluation methods and found that the student satisfaction feedbacks from the evaluation help point out the desirable and preferable teaching methods. For example, class discussion activities which are within the classroom environment or outside, use of case students and multimedia in the classroom have attracted students' satisfaction and provided the enhancement to their learning (Chalmers, 2008). Other results of the evaluation show that students' academic success relies on certain features of learning environments, notably on small-group work and problem-solving exercises (Militaru et al., 2015). Also the integration of new concepts and ideas by teachers or lectures rather than just relying on the given one source of information like lectures has resulted in improving the overall experience of the students and their level of satisfaction as they mentioned in the given evaluation. Another evaluation found that blended learning with well-designed content and orientations has proven to be a good solution for improving student satisfaction with interaction in virtual environments (Chang, 2013).

It is established that academic institutions prefer to construct well-aligned study modules to enhance Student Satisfaction. These modules include one or the other of the afore-mentioned learning activities. But how do the students judge the well aligned study modules? A questionnaire conducted by the "National Student Survey", University of Birmingham (2012), which focused on many questions describing what students liked about the module and why. The response of the students ranged from 'excellent' to 'good'. It gives an insight into the students' perspective of the well-aligned modules. Therefore, with data on student satisfaction, we can investigate and identify what attracts student satisfaction in good teaching practices thus it can be considered as health indicators to validate our alignment system.

2.9 Summary

The chapter reviewed existing learning design tools and concludes with their limitation in the absence of a metric system that measures the quality of good designs and enables intelligent design decisions to be made not only theoretically according to pedagogical theories but also practically based on good design principles. The computational framework of Tepper's alignment metric (2006) was reviewed and has shown that the level of constructive alignment can be measured numerically using vectorial representations and computations. The next chapter will initially present the first software implementation of measuring the degree of alignment. It is developed as a proof-of-concept prototype of Tepper's alignment metric where it is integrated with the LDSE in order to measure and visualise the degree to which an educational design is constructively aligned.

CHAPTER 3: Alignment Metric Engine: Design and Implementation

3.1 Overview

To address the measurement problem presented in the previous chapter a theory-aware alignment metric tool has been developed based on the simple mathematical model by Tepper (2006). The tool aims to measure how well aligned the module's components are when linked together and facilitate teaching practitioners to systematically and consistently produce constructively aligned modules of teaching and learning based on the design principle of constructive alignment (Biggs 2000). For each educational design, the metric tool is able to generate three different types of tree structure, measure the alignment at individual tree and across tree levels, and recommend/suggest more appropriate objectives, TLAs, and ATs to improve alignment. To do so, the metric tool takes as input an XML design pattern and maps the design components to their associated level in Bloom's Taxonomy. The Bloom levels of the components are then fed into the alignment metric engine to compute and measure how constructively aligned the components are and therefore aid the designer in making alternative, better-suited, selections. The computational method uses set theory and simple linear algebra operations to express, represent and compute alignment (Tepper 2006). The MySQL database system has been utilized to implement and manage the data model of the educational design (i.e. the learning outcome and objective verbs, TLAs, ATs, and their respective levels in the Bloom taxonomy. Queries to input or retrieve information from the database are expressed in the industry standard database language, Structured Query Language (SQL)[10]. The database design and implementation activities of the project considers aspects relating to the design of the user interface and output reports and the integration with the overall structure of the metric tool, and the computation of the alignment metric. Finally, the existing learning and design tools reviewed in chapter two informed the selection and design of the functionalities of the metric tool.

[10] http://www.sqlcourse.com/intro.html

3.2 Conceptual Schema of the Metric Engine

The computational framework behind the metric engine was given in detail in Chapter 2. In order to develop the data model, the data structure from the LDSE XML files was analytically reviewed. Subsequently, the conceptual schema illustrated in Figure [3.1] was developed to provide a high level representation of the underlying normalised data model and thus database tables and columns.

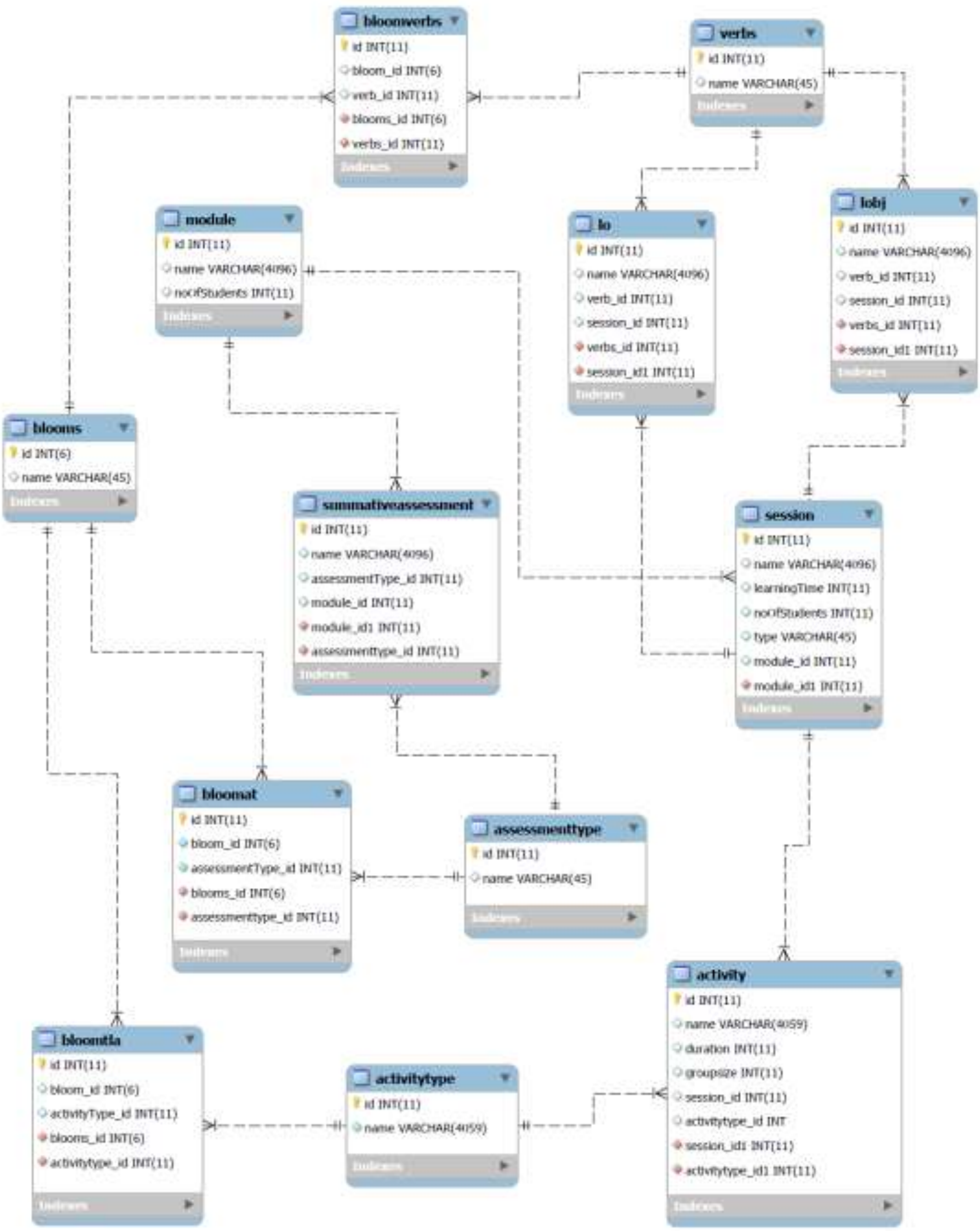


Figure [3.1]: The conceptual model of the alignment metric engine

3.3 Database Development

The chosen database management system to store and analyse the educational components was SQLite Manager due to it being a free open source database management system, compatible with many programming languages, and easy to implement which makes it suitable for applications such as the model prototype being developed for this research project. It also provides a Graphical User Interface (GUI) that is easy and efficient to use. Multiple tables were therefore created to store the different components or attributes about the module designs. For each component, the respective levels from Bloom's Taxonomy were then stored in the database across the four tables; the six categories, their level and definition were stored in the Bloom's_Taxonomy Table and then the verbs and their associated category were stored in the Verbs_Taxonomy table, the TLA types and their associated category were stored in the TLA_Taxonomy table, and lastly the assessment types and their associated category in the AT_Taxonomy table. Referencing between these tables occurred with the use of primary and foreign keys. The use of the database tables allowed for given information to be broken down, stored separately and then analysed through the construction of the associated Bloom's level. It is worth noting at this point that the verbs, TLAs, and ATs stored in the database are not an exhaustive list but ones which are based on LDSE and cannot use verbs, TLAs, or ATs beyond those recognised by the LDSE as we want it to be compatible with the LDSE and therefore extend its functionality based on theory.

Due to reason that there are no openly available data models for the LDSE, therefore; there is a need to deduce the data structure analytically from the exported LDSE XML files. The following analysis was performed:

- Identified key learning and teaching concepts represented from the exported XML file to produce appropriate fields of data for the database tables.
- Identified relationships between concepts and how these were structured. The structure of the design pattern data of the LDSE is outlined in Table [3.1] below. (*The full xml structure is illustrated in* **Appendix [A]**).
- Transformed data into third normal form (**Normalisation[11])**
- Implemented the set of normalised tables using MySQL database system.

[11] Normalisation is process that prepares a data model for implementation as a simple, non-redundant, flexible and adaptable database. http://www.1keydata.com/database-normalization/third-normal-form-3nf.php

- Adopted a reverse engineering approach to generate an Entity-Relationship Diagram (ERD) from the implemented MySQL tables.
- Expanded the resulting ERD to include additional entities holding the necessary data and information to be ready for alignment processes.

Table [3.1]: XML Structure of LDSE Design Pattern

Module details	<module elapsedTime="0" name="New unit 1" noOfStudents="0" staffTimeAllocated="0">
Module's assessment	<summativeAssessment/>
Session details	<session elapsedTime="0" learningLevel="1" learningTime="180" name="Teacher Supported Class F E" noOfStudents="15" scheduleEnd="1262314800000" scheduleStart="1262304000000" topic="" typeID=" TeacherSupportedClassFE "> <id>1380245462698</id>
Session's outcomes	<outcomes> <outcome verb=" "><![CDATA[]]></outcome>
Session's activities	<activity description="" duration="30" groupsize="5" name="Group Practical Activity" notes="" start="1262304000000"> <teachingmethod group=" TeacherSupportedClassFE " id="DefaultGroupPracticalActivity" name="Group Practical Activity"/> </activity>

3.4 Mapping LOs, LObjs, TLAs, and ATs to Bloom's Taxonomy

The different verbs, TLAs and ATs collected from LDSE were stored and assigned to their appropriate cognitive process dimension according to Bloom's Taxonomy as it is a familiar tool to most teaching practitioners. The action verb is the key element in stating the learning outcomes and learning objectives that defines student learning according to Bloom's classification. The level of the each learning outcome and objective is therefore linked to an associated level in Bloom's taxonomy based on the main active verb by matching the main verb with the corresponding entry in Bloom's taxonomy that contains the matching or synonymous verb. The level of Bloom's assigned for each TLA and AT is based on the type of learning they elicit or assess as provided in Biggs (1999) and (2003). This is not a precise grouping however; Tepper (2006) has adapted this approach to link TLAs and ATs with particular levels in Bloom's Taxonomy to help clarify the alignment computation process. The alignment tables are provided in **Appendix [B]**. Matters became more complicated when verbs, TLAs, and ATs were found in more than one level or category of the taxonomy. This was resolved by referring to multiple tables and taxonomies defined by educational institutes Anderson (2001); Almerico and Baker (2004); Biggs (2003); Biggs and Tang (2007) to determine which one it was best suited. For the purpose of this research and in order to simplify the classification and aid in the analysis process, each verb, TLA, and AT was classified into only one of the six Bloom's categories where index to the highest level in Bloom's Taxonomy is taken.

3.5 Functional Requirements

The conceptual model as seen is represented by the knowledge base, which simply requires that the system components can be categorised according to the cognitive ability they elicit and on this basis make dependency relations across component groups to form structure. The metric tool has been designed to augment the LDSE with new information based on the principle of constructive alignment. Thus, some high level functional requirements of the metric tool have been identified below and graphically illustrated in Figure [3.2].

1. Import an XML design pattern that is exported from the LDSE to be read and analysed regarding the four main learning components; LOs, LObjs, TLAs, and ATs.
2. Store data of the XML design pattern into database tables. This function will be used to store data and to assign Bloom's level to each component. Data are retrieved when the alignment computation is performed.
3. Generate the alignment trees:
    a. Generate Outcome/Objectives perspective tree. This function is presented to the user to enable the user to link or associate the learning outcomes identified in his/her module or session with the learning objectives, generating a perspective tree that clarifies each outcome and its dominated objectives along with the cognitive skills each component elicits.
    b. Generate Objective/TLAs perspective tree.
    c. Generate AT/LO perspective tree.
4. Calculate alignment between components:
    a. Calculate alignment between outcome and objectives. This function will perform mathematical operations based on the alignment metric.
    b. Calculate alignment between objective and teaching activities
    c. Calculate alignment between outcome and assessment tasks
5. Generate an alignment Report. After analyzing the components and calculating the misalignment error value, the user will be able to generate an alignment report showing whether the relation between components is aligned or misaligned.
6. Modify misaligned components to produce a more aligned design by suggesting more appropriate verbs, activities, or assessment tasks for the user to consider.
7. Calculate the overall module or the design pattern alignment. This will calculate the alignment between the generated trees (design components) in order to represent the overall alignment to the user or the designer.
8. Export the design pattern.

These functional requirements of the alignment metric tool are designed to aid teaching practitioners in identifying and relating the core design components symbolically through a graphical user interface whilst abstracting them away from the actual alignment computations used to determine the alignment measures. The requirements were modelled using the concept of use case diagrams, which was initially introduced by Ivar Jacobson in 1987, to describe the sequences of actions and illustrate the user's interaction with the system. Their benefits lay in showing the relationship between the user and the different use cases in which the user is involved. They also serve as an easily-understood communication mechanism and provide a concise summary of what the system will do at an abstract level.

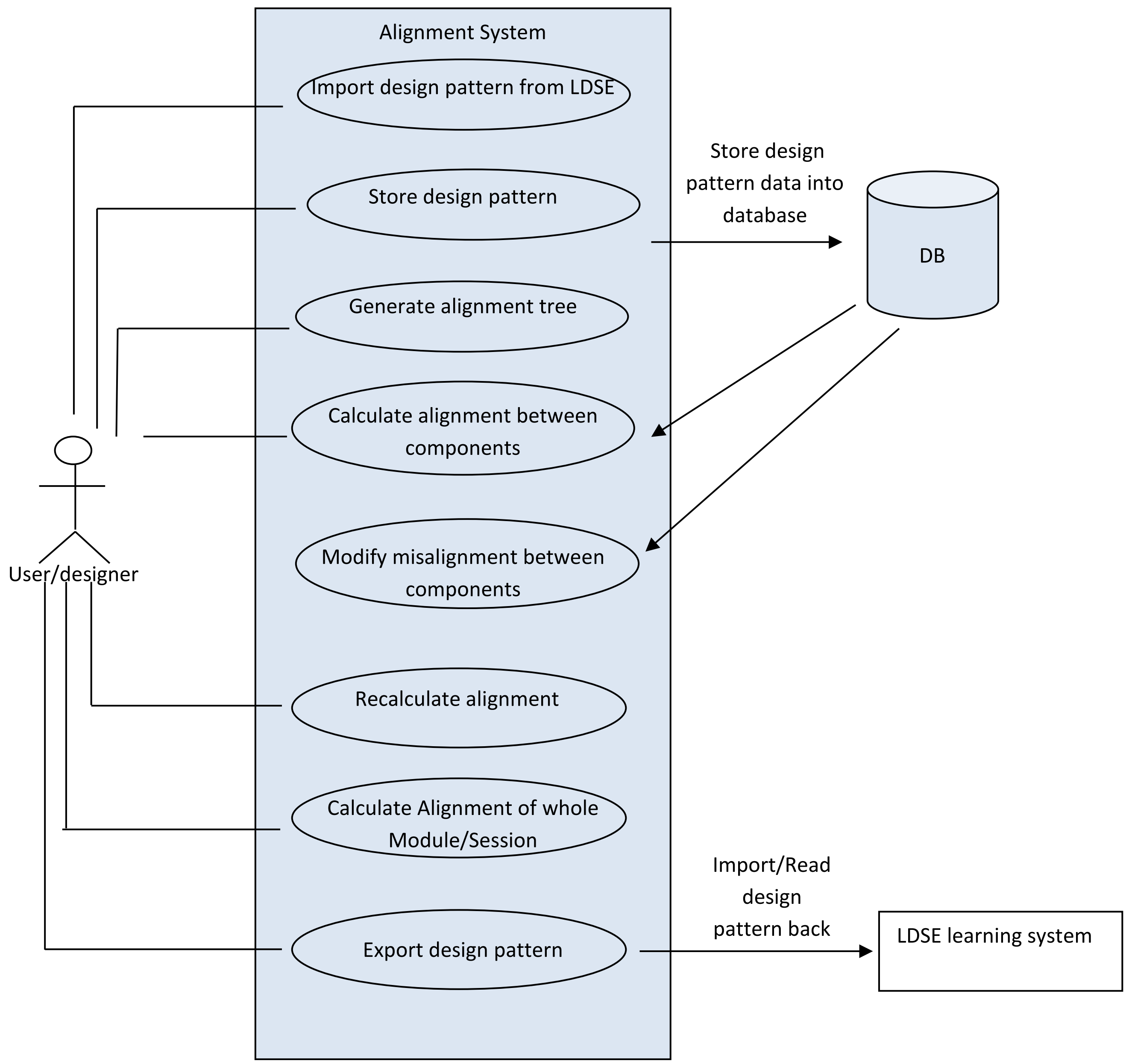


Figure [3.2]: Use case diagram illustrating the functional requirements of the Alignment Tool

3.6 Implementation of the Alignment Metric Tool for Measuring Design Quality

The implementation of the alignment metric engine and user interfaces is achieved in C# with Microsoft Visual Studio 2010, which aimed to show the high level functionalities of the metric engine, performance and facilities. The C# programming language is simple, powerful, type-safe, and object-oriented that is designed for building a variety of applications that run on the .NET Framework. The .NET Framework includes an extensive library of classes organized into namespaces that provide a wide variety of useful functionality. In addition, it gives powerful tools for creating, loading, and saving XML files.

Figure [3.3] provides an illustration of the main components of the alignment tool. The conceptual model is represented by the learning design database and includes the relationships between core module components and the cognitive ability they elicit that support the design principle of constructive alignment. The database also serves the data layer and provides a couple of operations to work with XML files. The alignment engine at this stage is supported by executable rules based on the principles of constructive alignment. On this basis, it makes dependency relations across component groups to form the tree structures. The user interface is how the user interacts with the alignment tool. It is made up of several window forms to interact with. Figure [3.4] shows the main interface window that contains the main tool's functionalities. Each window relates to a key aspect of functionality which will be described in more detail with some screenshots.

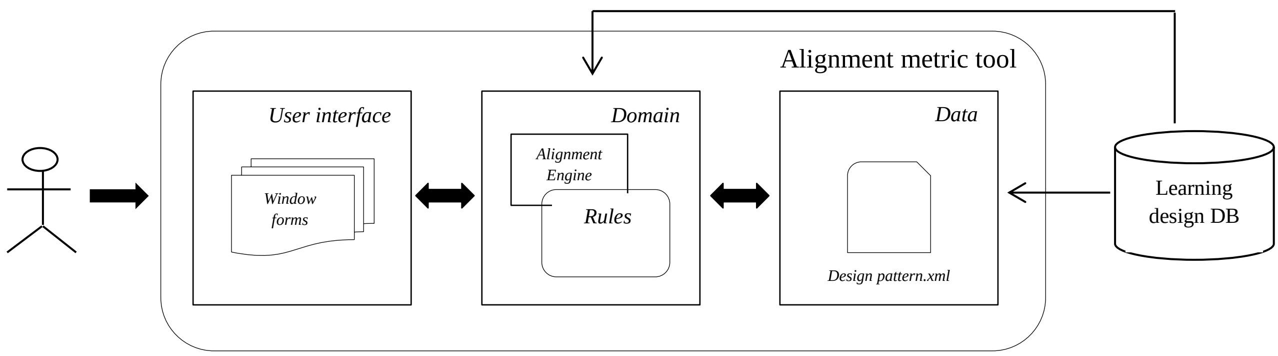


Figure [3.3]: Overview of the alignment tool architecture

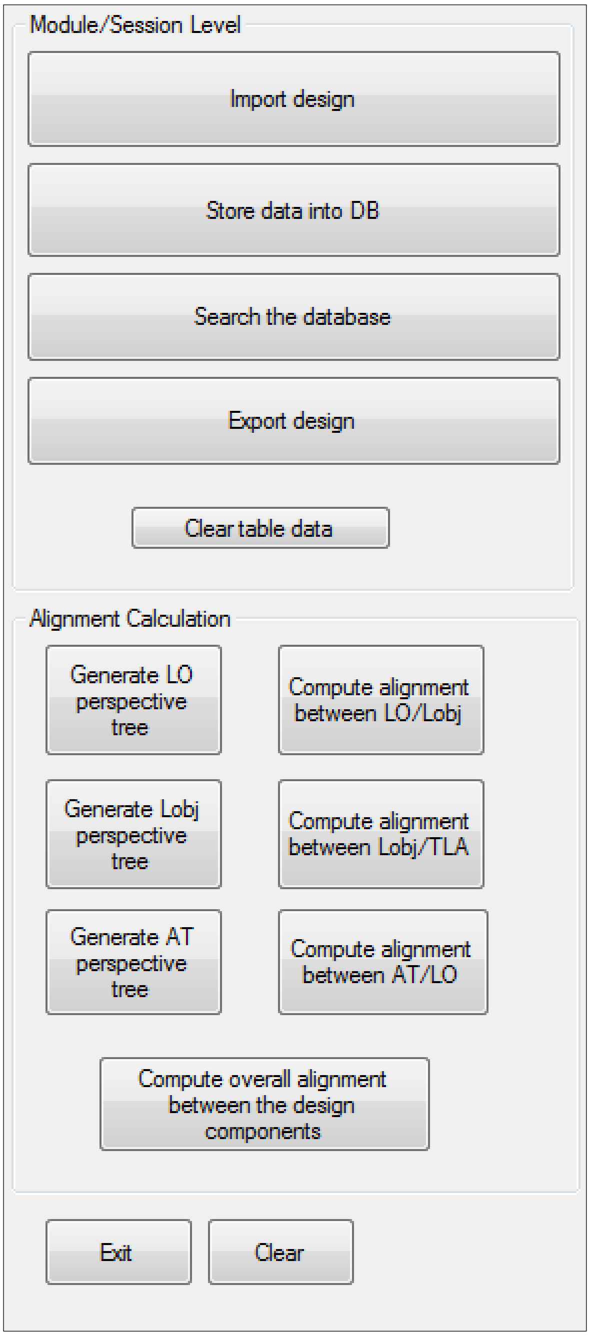


Figure [3.4]: The Alignment metric tool's main functionalities

Step 1- Import LDSE learning design pattern

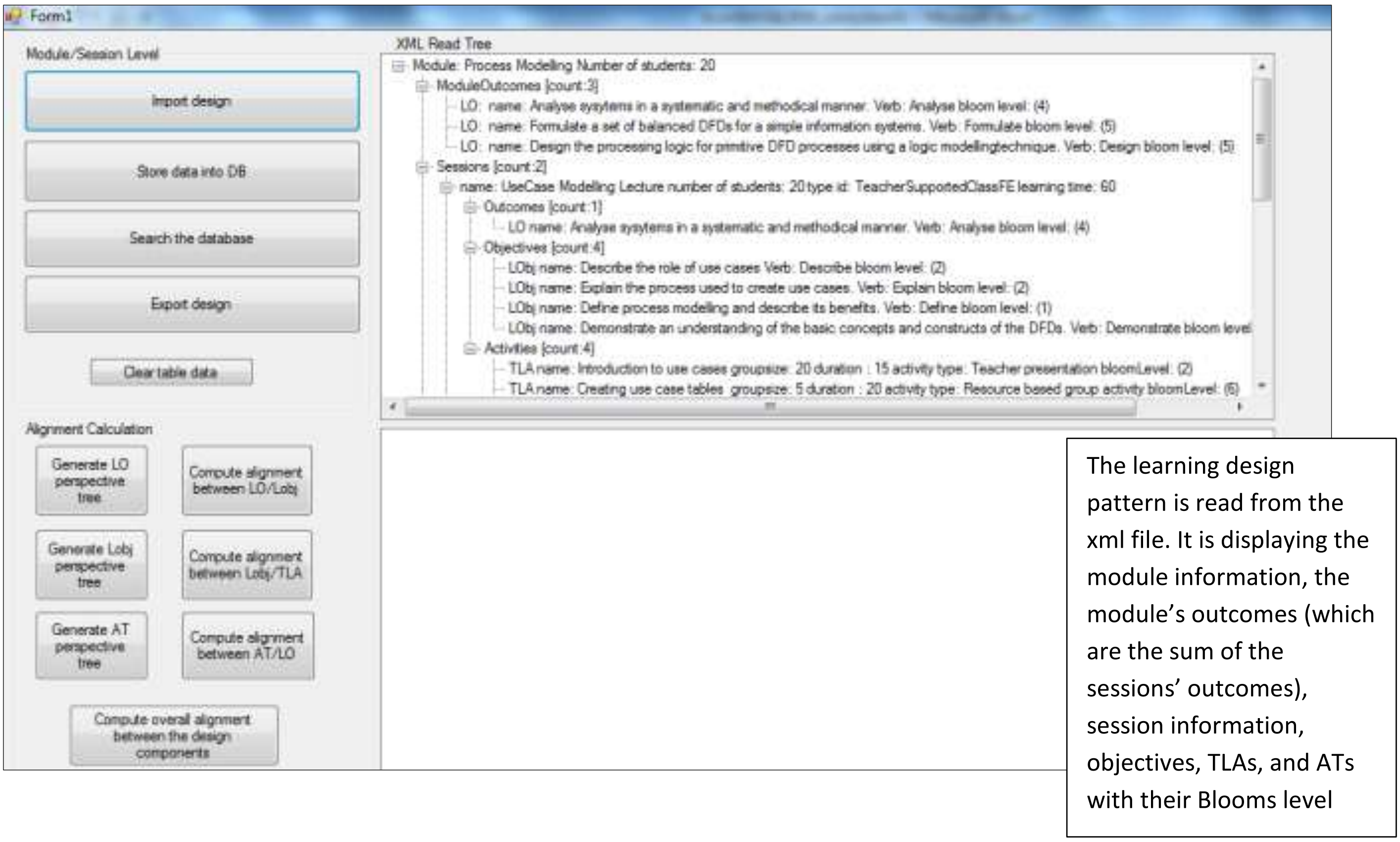


Figure [3.5]: The 'Import learning design' window

In the first step the user needs to click the import button in order to import his/her learning design pattern. Clicking the button will open a dialog file to select the design pattern. The model reads all the required data from the design pattern (XML file) and displays the following components: learning outcomes, learning objectives, TLAs, and ATs of the module with the associated Bloom's level for each component. The Bloom levels are coming from the knowledge stored in the database.

Step 2- Generate LO perspective tree

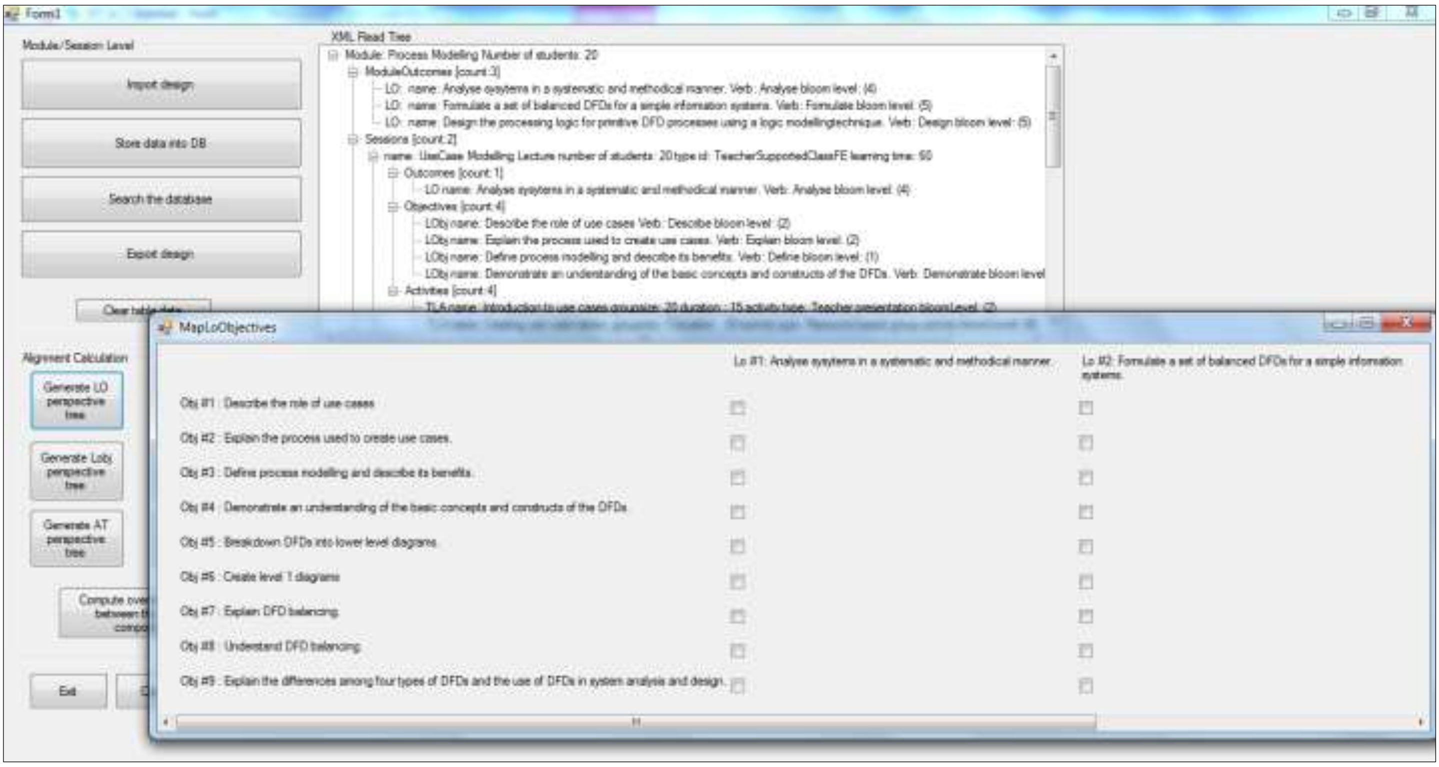


Figure [3.6]: Checkbox dialog window to generate the learning outcome perspective tree

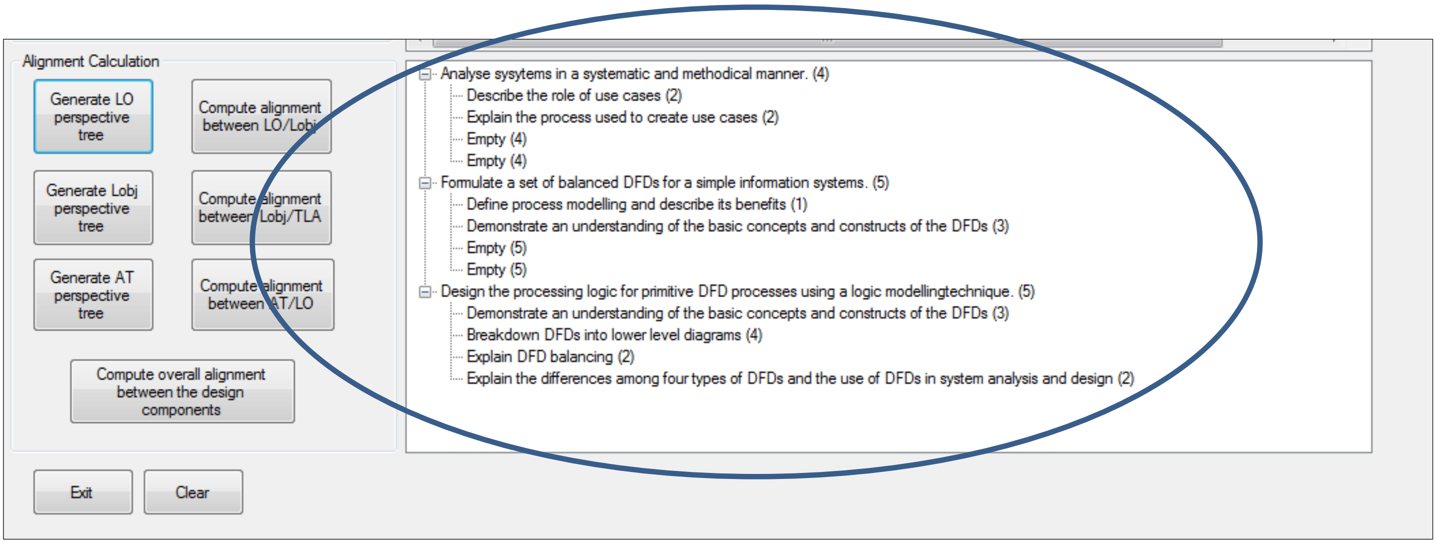


Figure [3.7]: Generating LOs/Lobjs tree structure showing relationships between outcomes and objectives

The user here needs to generate a LO/Lobj tree to calculate the balance between these two components. A dialog box will be displayed when the button “Generate LO tree” is clicked. The dialog box will display all the learning outcomes and the learning objectives identified in the module and enable the user to link the learning outcomes with the associated learning objectives. One or more learning objectives can be associated with the learning outcome as in Figure [3.6]. When the user links the components together and presses the button ‘submit’, the generated tree is displayed showing each learning outcome and the dominated learning objectives. An Empty node is added to the tree to balance the number of dominated elements per outcome. The empty node takes value set to the level in Bloom’s indexed by the associated outcome as seen in Figure [3.7] above.

Step 3- Calculate the alignment between Lo/Lobjs

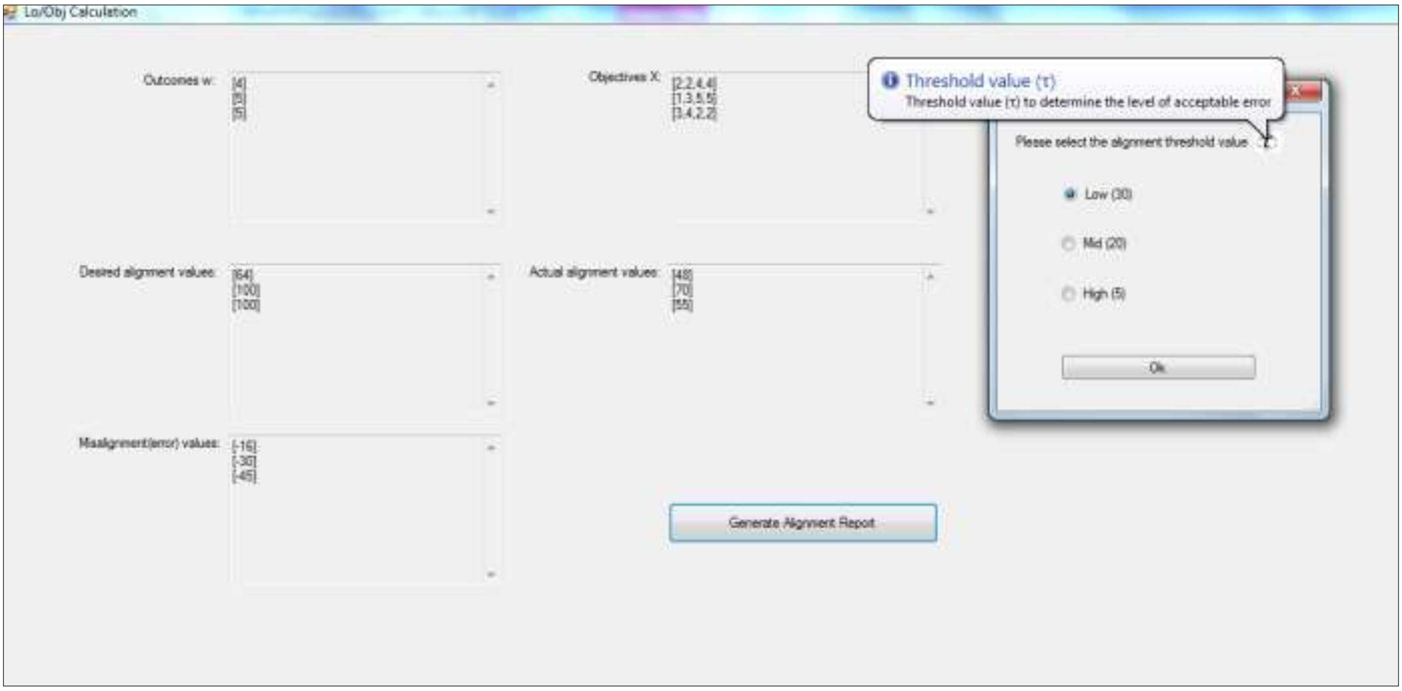


Figure [3.8]: The ‘Alignment calculation’ window form to calculate the alignment between the learning outcome and its dominated learning objectives

The screenshot illustrates the Alignment calculation according to Tepper (2006). Once the LO tree is generated as in Figure [3.7] the user can computes the alignment between the learning outcomes and learning objectives. The calculation is performed by calculating the following: 1-actual alignment values using the inner dot product between each learning outcome and its corresponding set of actual learning objectives stored in matrix X. 2-desired alignment values using the inner dot product between each learning outcome and its corresponding set of desired learning objectives. The crude assumption made to obtain the desired elements is that given a learning outcome the set of associated learning objectives should elicit the same Bloom's level as the learning outcome. 3-misalignment error values between the desired alignment values and the actual alignment values for each learning outcome. Then for each learning outcome the tool compares the misalignment error values to the threshold value that been set using the illustrated piece of structured English and thus generates an alignment report as seen in Figure [3.9].

For each learning outcome i
Do
If |e1i|<=τ
Then If one or more x'ji = wi' (*for each j*)
    Then the learning objectives are *aligned* with learning outcome i
Else
Learning objectives are not fully aligned with learning outcome i
Else If |**e1**i|>τ AND **e1**i >0
    Then If one or more x'ji = wi' (*for each j*)
    Then the learning objectives are *positively misaligned* with learning outcome i
Else
The learning objectives are not fully aligned with learning outcome i
Else
The learning objectives are *negatively misaligned* with learning outcome i

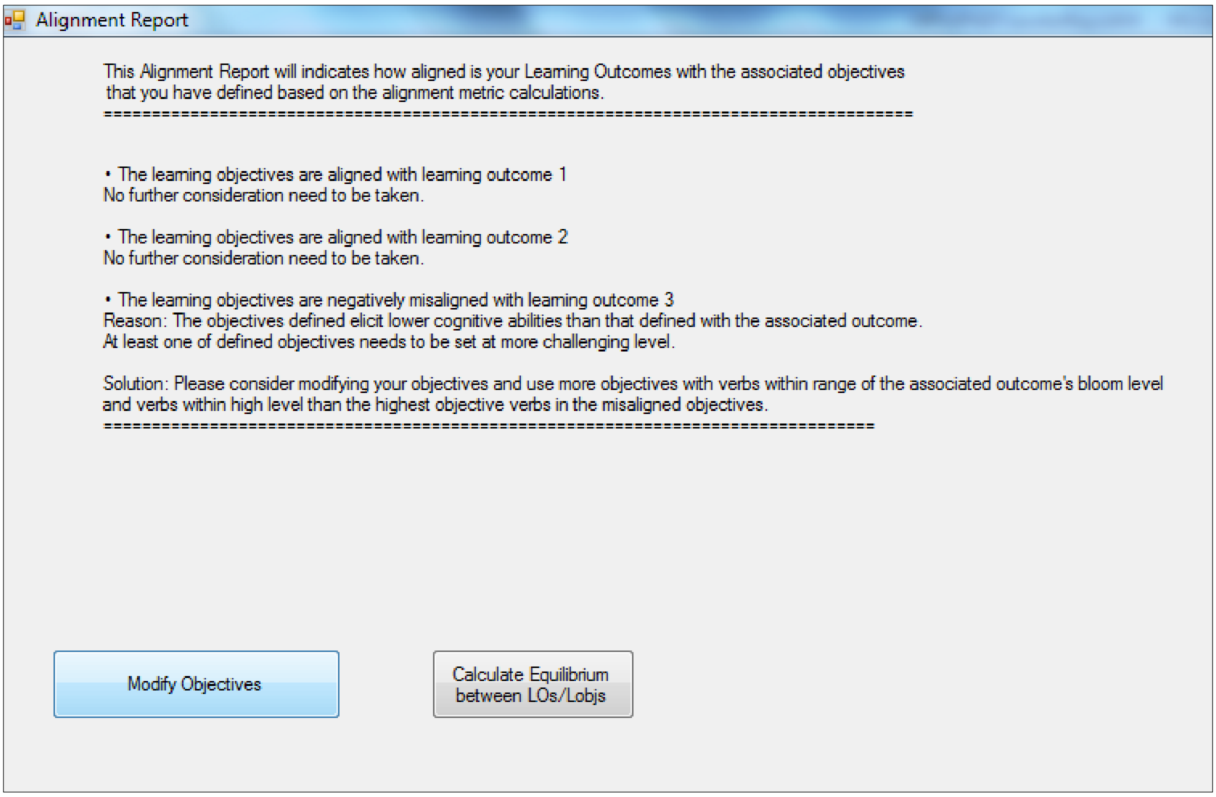


Figure [3.9]: Generating the Alignment report

The alignment report illustrates the aligned and misaligned elements based on the alignment metric calculations. As shown from the report some learning objectives are negatively misaligned with the associated learning outcome because the user has defined learning objectives that elicit lower Bloom's level than that defined in the associated learning outcomes' Bloom level. This means even if the learners achieve all the learning objectives the learning outcome is still not achieved. In this case the user has the option to carry on and calculate the equilibrium or to modify the misalignment in the design pattern. If the button 'Modify Objectives' is selected, then the next step is presented to the user.

Step 4- Modify the misalignment

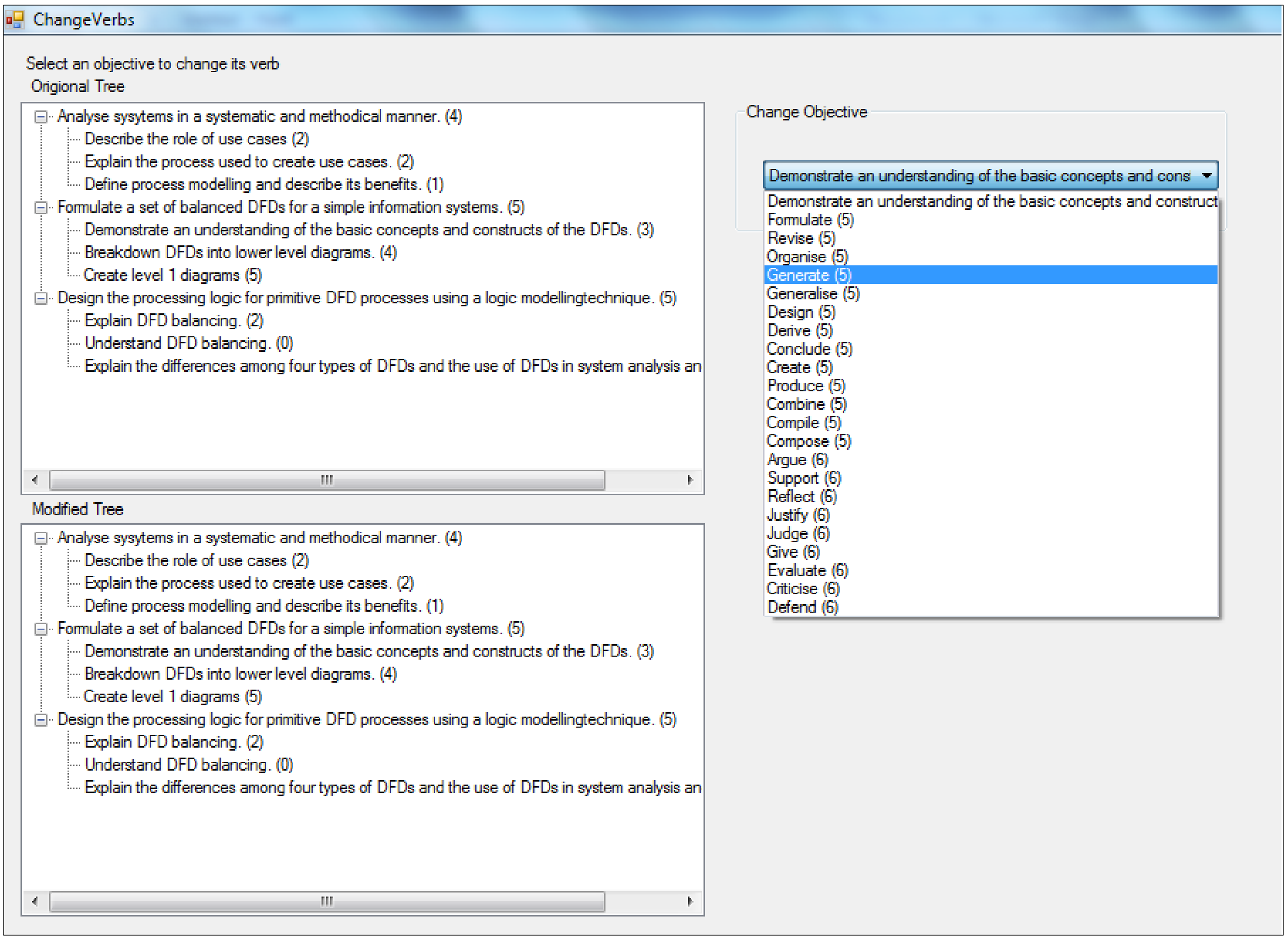


Figure [3.10]: The ‘Modify misalignment’ window form

The model allows the user to modify the misalignment components by displaying a change window form that contains two tree view controls. The misalignment tree is cloned inside both tree controls, one has the original tree and the other one has the modified tree to allow the user to track the changes. The model enquires the database for verbs that have Bloom levels within or greater than the original Bloom level of the parent’s node (i.e. outcomes) and displays these suggestions in a drop down box to select from. The user’s changes will be saved, reloaded, and recalculated again. The model will be notified that some changes have been made to the tree and thus re-balance the tree and re-calculate the alignment between components. It gives the user the opportunity to reconsider the choices and make more aligned decisions.

Step 5- Calculate the equilibrium

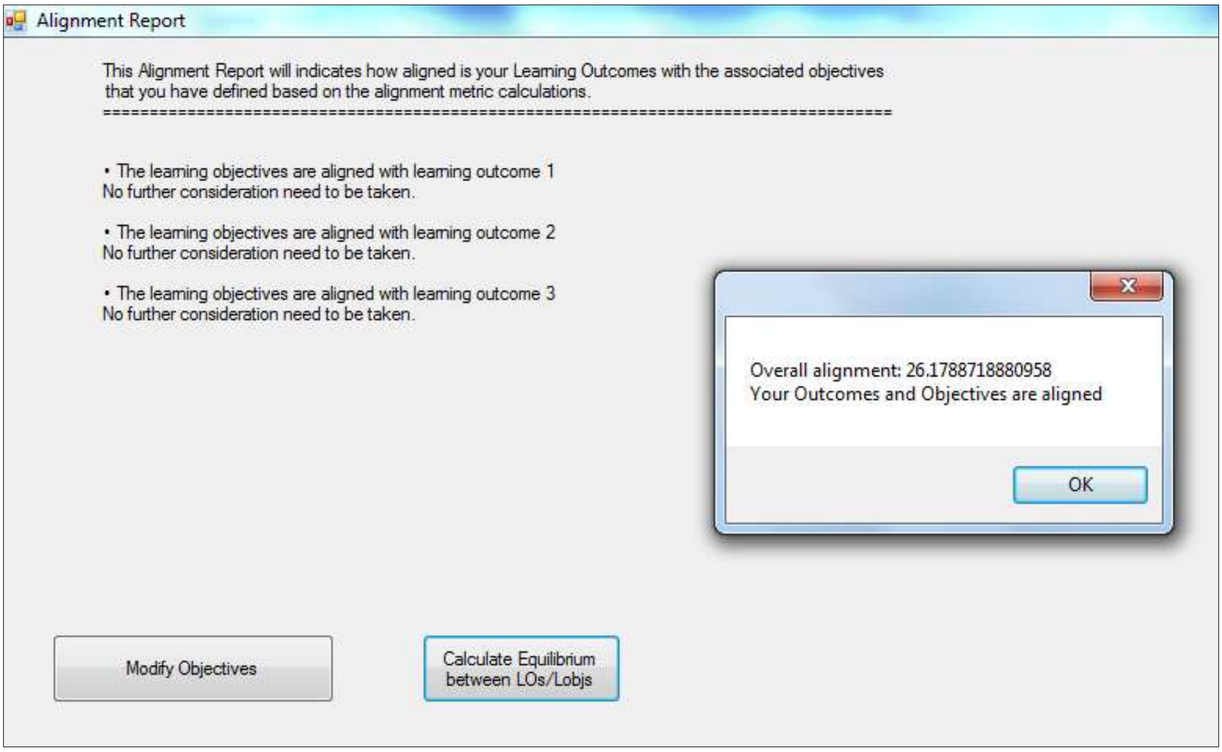


Figure [3.11]: An overall Alignment score between components

This screenshot shows an alignment report again after modifying the misalignment and re-calculating the alignment. The five steps are applied again for the other components (the learning objectives and activities and the assessment tasks and the learning outcomes). Then the overall alignment consolidating all components and representing the overall constructive alignment by adding the equilibrium output from each tree performing the equation:

$$\frac{\text{LOTree.equilibrium} + \text{ObjTree.equilibrium} + \text{ATTree.equilibrium}}{3}$$

3.7 Adequacy of the Alignment Metric Tool for Measuring the Design Quality

The developed alignment tool can measure the degree of alignment between individual system components and of full constructive alignment for an entire module using victorial representations and mathematical computations. It has been augmented with the LDSE learning design system to analyse a number of module design patterns. This has resulted in a novel and much needed enhancement to the LDSE in that the alignment tool enables LDSE design patterns to be objectively measured by computing the degree to which a module is constructively aligned. It enables users to measure their design quality, to visualize alignment, and to modify the design patterns to further improve alignment scores of their designs. The tool is not intended to replace existing learning design tools; however, its aim to offer quantitative measure of alignment through an easy and accessible system that can aid both specialists and non-specialists. Conole mentions that, "*the development of toolkits provides a way for non-specialists to engage with such theories in a manner which supports careful design and prompts productive reflection and engagemen*t" (Conole, 2004).

The developed alignment tool has been evaluated on a number of module design patterns to measure the degree of alignment and to constructively align the misalignment components. Figure [3.11] presents an example of four selected module designs with different learning outcomes and the other associated components. The examples are selected randomly and for the purpose of demonstration. The alignment is computed between the different components and the overall module alignment is then given under each module to show the degree of alignment or misalignment each module presents. It is noticeable that there is no balance between the components leading to degrees of negative misalignment in some modules like 2, 3, and 4. These negative misalignments are indicators of poor design due to first, learning outcomes are not linked with an appropriate Bloom's level of learning objectives. Second the TLAs used in these modules are not designed to generate or elicit the desire verb of the associated learning objectives. Finally, it can be seen that there is little association matching taking place between the learning outcomes and the assessment tasks. The tool attempts to represent the alignment/misalignment numerically with a brief description of the relationship between the components. Alternative LObjs, TLAs, and ATs are recommended during the design process to elicit the same or higher cognitive level than the associated parent node in order to maximize the alignment as shown in Figure [3.13].

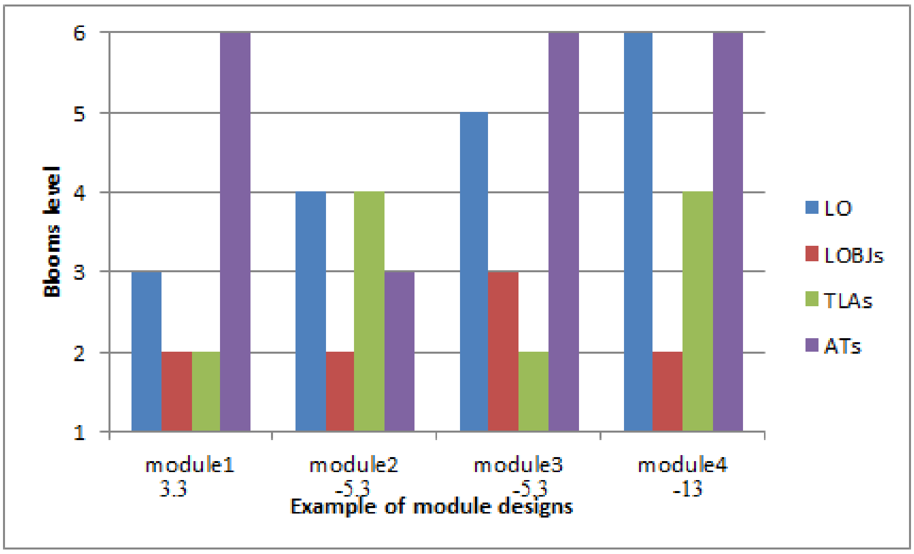


Figure [3.12]: Example of misaligned module designs

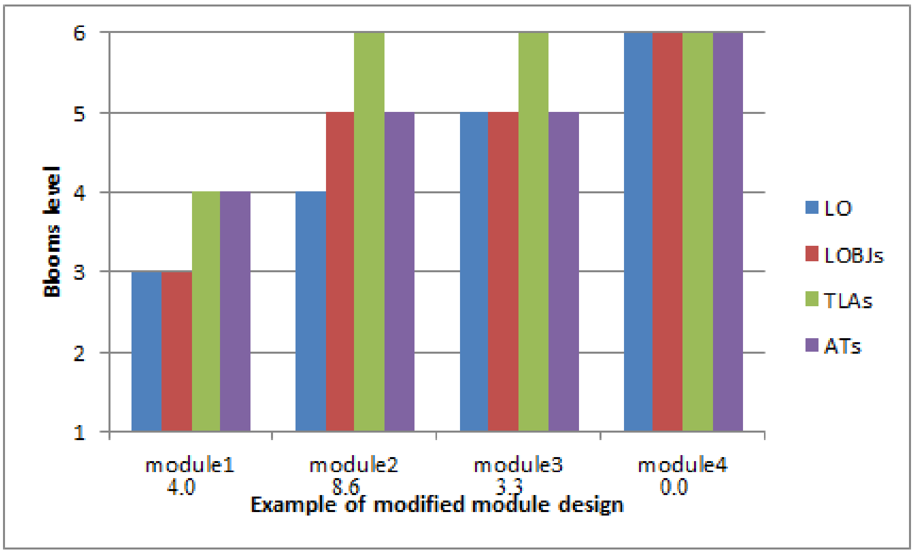


Figure [3.13]: Example of modified modules

The evaluation also found that there is a high proportion of misalignment taking place in particular between the learning outcome and the assessment tasks. Following that with 33% of misalignment usually occurs between the learning outcome and learning objectives as illustrated by Figure [3.14]. This is likely because most teaching practitioners find the distinction between learning outcomes and learning objectives somehow nebulous, therefore find it difficult to relate the two. D'Andrea's (1999) makes it clear that one is the output (learning outcomes) and the other is the input (learning objectives) to the TLA process. The learning outcomes determine the list of learning objectives that the students are required to achieve in order to attain the outcomes. The objectives subsequently determine the teaching and learning activities that teaching practitioners need to apply to engage students.

The impact of the alignment tool will help teaching practitioners to maximize alignment by enabling them to adapt their practice to better align their modules and making them aware of misalignments within their educational designs. Thus allowing them to infer and make the appropriate associations between the core educational components, which will positively enhance the students learning and satisfaction.

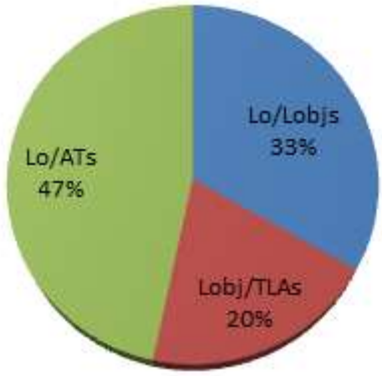


Figure [3.14]: Misalignment proportion percentage

3.8 Limitation of the Alignment Metric Engine and Possible Solutions

The developed alignment metric bases both module structure and the categorisations of the different components on pedagogic theory alone (i.e. outcomes-based approach to module design, components are organised according to Bloom's taxonomy and related according to the principles of constructive alignment). This is a top down theoretical approach to educational design and there is no clear consensus as to what the appropriate alignment values should be. It is important for the alignment system to be cognisant of this in order to offer pragmatic and realistic design solutions. Therefore, the alignment metric engine needs to be extended to incorporate and use good design practices, as judged by student satisfaction scores, to calibrate the alignment measures and thus determine acceptable alignment ranges based on effective practice. Examples of effective practice, as judged by student satisfactions rather than theory in and of itself will be used to identify allowable and acceptable values for the alignment threshold, thus integrating both theory and practice into its decision making. Reviewing good teaching practices that generate high levels of student satisfaction may help to provide useful insight into the value of theoretical alignment values, particularly if there appears to be a strong correlation between the two i.e. alignment scores and levels of student satisfaction. An advantage of this approach will be to help teaching practitioners to better align their module designs in a way that is both theoretically and practically relevant and based on those actually experiencing the impact of the educational design. This is a new research itself as no such metric system exists to date that investigates the linking between theory and effective design practice. Moreover, in order to introduce an adaptive knowledge base for the alignment metric that can learn alignment information directly from the good module design patterns, the use of artificial neural networks will be used as a novel approach for adaptively supporting the educational design process in a way that marries constructive alignment with good design practice and therefore incorporate both theoretical and practice-based models of design practice. Figure [3.15] illustrates an overview of the proposed modular architecture.

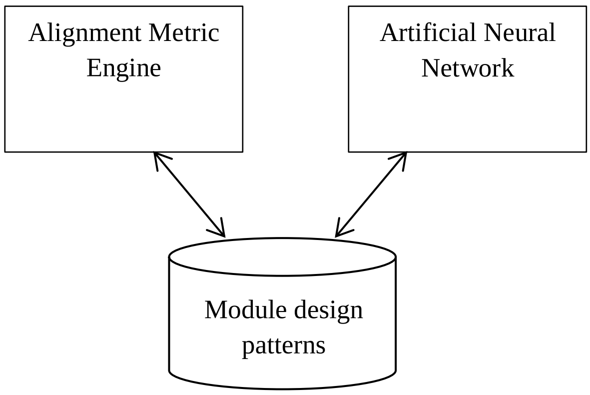


Figure [3.15]: Modular architecture

3.9 Artificial Neural Network: A Possible Solution

Neural networks are adaptive systems that have learning properties enabling them to adapt their internal parameters in order to satisfy constraints imposed by a training algorithm and by the input and output training data. Instead of following a set of rules specified by a human expert, neural networks are universal function approximations that can learn any function and learn the underlying rules and input-output relation from collection of representative examples. They have the ability to learn and remember portability, to distinguish objects, and to make intelligent decisions (Jain et al., 1996). Neural networks, with their incredible ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. They have the advantage of adaptive learning to learn how to perform tasks based on the data given for learning. Also they have been described as self-organisation as they can generate their own representation of the information received during learning time (Christos, and Dimitrios (2015). Their usage has become very widely used in many applications such as classification, pattern recognition, feature extraction, image matching, forecasting, and data compression, data clustering, optimization, pattern completion, and associative memories.

Pattern completion is the ability to recall a stored representation when presented with a partial or corrupted observation of the stimulus. In referring back to Hanson and Kegl (1987), they have used neural networks and in particular, auto-associative neural networks (AANN), to perform sentence completion on sentence fragments, prefer syntactically correct sentences, and to recognize novel sentence patterns absent from the presented corpus. Their work is quite similar to this research approach in applying the same principles, however; the approach will be applied on module design patterns. In so doing, the network will memorise module designs that yield high-levels of student satisfaction (and related alignment metrics) to the required degree of accuracy by forming a compressed set of hidden unit representations (features). After successful learning, the AANN will be presented with novel test patterns where the expectation will be that the AAN will attempt to match any novel or new input patterns to those it had learnt during training. These test patterns consist of input patterns from those modules with low student satisfaction scores. When presented with such test patterns, the AANN will effectively treat these as noisy versions of patterns within the training set and therefore attempt to produce a pattern on the output layer that resembles one or more (i.e. an aggregation) of the ‘good’ module designs found within the training set which is closest to the current input pattern – effectively using the features of 'good' module designs to identify changes to those module designs which have much lower student satisfaction. The advantage of using AANN for this task will help to extract regularities or discover some patterns within the data that are useful in predicting the output stimuli. Another advantage is as the networks learn by changing their behaviour; this makes them perform better in the future by introducing self-sustaining and adaptivity by accepting both new patterns as input (which have high student satisfaction scores) and input patterns which it has generated on the output to extract new module designs. The next subsections will explain generally some of the theory parts of artificial neural network describing the different types of auto-encoder architectures.

3.9.1 Artificial Neural Network Overview

An Artificial Neural Network (ANN) is a computational model that is inspired by the way biological nervous systems, such as the brain, process information. A boarder definition of neural network is captured by Samarasinghe as "*a collection of interconnected neurons that incrementally learn from their environment (data) to capture essential linear and nonlinear trends in complex data, so that it provides reliable predictions for new situation containing even noisy and partial information"(* Samarasinghe, 2007*).*

- **Architecture of Neural Networks**

Neural networks can be categorised into two main categories:

1) Feed-Forward Networks in which no loops are formed
2) Feedback Networks in which loops occur because of the feedback connection.

Feed-forward networks are one of the most common used network architecture that allow signals only travel one way from input to output and no loops are formed by the networks. Learning in Feed-forward ANNs uses a *supervised* learning algorithm, in which both input and output patterns are known and presented to the network so that the network 'learns' the relation between the input and the output (Samarasinghe, 2007). The most commonly used family of feed-forward networks are the multilayer perceptron (MLPs) networks in which neurons are organized into layers with connections strictly in one direction from one layer to another (Jain et al., 1996). Neural networks can be defined based on three main characteristics as mentioned in (Swain et al., 2012) these are:

1- The architecture including the number of layers and number of nodes.
2- Algorithm mechanism applied for updating the weight of the connection.
3- The activation function.

- **Network Layers**

Neural networks generally consist of three types of layers: input layer, output layer, and one or more hidden layer. The input layer accepts the input information from the outside environment and sends them to the hidden layer. The activity of the hidden layer(s) is to calculate the weighted sum of the inputs which is then passed through linear/non-linear activation function to produce the outputs in the output layer which is heavily depends on the activity of the hidden units and the weights between the hidden and output units.

3.9.2 Auto-associative Neural Networks (AANN)

AANNs are feed-forward networks whose input and output vectors are identical. The process of training is called storing the vectors, which can be retrieved from distorted or corrupted input, if the input is sufficiently similar to it. AANNs are typically used for tasks involving pattern completion as stated earlier. The performance of the network is derived from its ability to reproduce a stored pattern from a corrupted input (Metcalfe, 1991; Weber and Murdock, 1989). The association in the network is achieved through the interaction of a set of simple processing elements, which are connected through weighted connections that can be adjusted in order to change the input/output behaviour. AANNs form a suitable approach for association rule mining as they store associations among the patterns. Thus output rules are extracted from a trained knowledge apposite to other approaches such as Apriori algorithm, which is array-based storage structure (Duch, Adamczak and Grabczewski, 2010; Setiono, 2011).

3.9.3 Multi-Layer Perceptron

A Multi-Layer Perceptron (MLP) is a feed-forward artificial neural network that consists of an input layer, one or more hidden layers and an output layer as shown in Figure [3.16]. The figure illustrates a simple MLP with one hidden layer as it has been proven mathematically and theoretically that MLP network with one hidden layer is capable of approximating any non-linear function to arbitrary levels of precision (Hornik, 1991), Kaastra and Boyd (1996), Bishop (1995). In order for the MLP to perform as an auto-association task, the input and output layer must have the same number of units or neurons where the number of hidden neurons in the hidden layer is less than the number of input neurons. The network is then trained to reconstruct its inputs, which forces the hidden layer to try to learn good representations of the inputs. The activation function f(x) for this task is a key component therefore; hidden neurons with non-linear activation functions are likely to be used to detect complex non-linear features. Typical choices for the activation function include the hyperbolic tangent sigmoid, with tanh(x) as in equation 1, which has been used in this research, or the logistic sigmoid function, with sigmoid(x) as in equation 2.

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

$$f(x) = \frac{1}{1+e^{-x}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

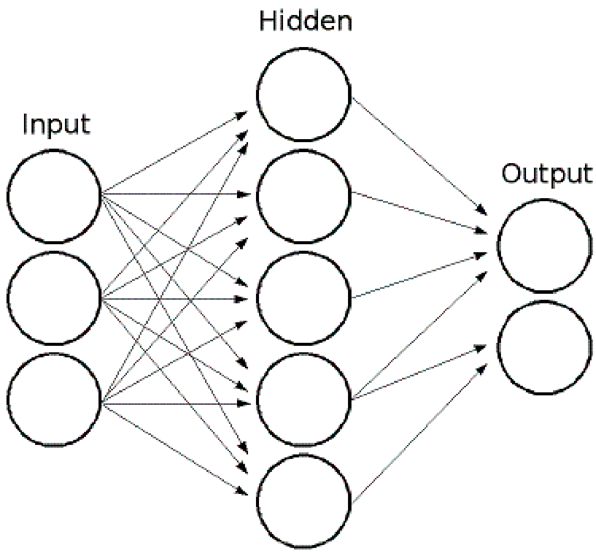


Figure [3.16]: Graph of a multi-layer perceptron with one hidden layer.

(Source: http://docs.opencv.org/2.4/modules/ml/doc/neural_networks.html)

Training a MLP is typically performed with a backpropagation learning algorithm, which is a common method of training artificial neural networks, used in conjunction with an optimization method such as gradient descent. The algorithm used to calculate the gradient of a cost function and bias values with respect to all weights in the network. The gradient is then used to find change in each weight and update it.

3.9.4 Auto-encoders

An auto-encoder, also called auto-associator or diabolo network, is an auto-associative neural network derived from the multi-layer perceptron which aims to recall and reconstruct their inputs into outputs with the least minimum error reconstruction (Bourlard and Kamp, 1988; Hinton and Zemel, 1995; Rumelhart et al., 1986). The aim of an auto-encoder is to learn a representation (encoding) for a set of data, typically for the purpose of dimensionality reduction. Architecturally, classical auto-encoders are simply FF-MLP with one or more hidden layers. Auto-encoders that consist of many hidden layers support deep network architectures that allow learning features from the datasets themselves. Figure [3.17] shows five layers as input layer, mapping (coding) layer, bottleneck layer, de-mapping (decoding) layer, and output layer. It is assumed that such layers—referred to as the bottleneck—compress the information needed for mapping the neural network input to the neural network output, increasing the system robustness to noise and over-fitting. The network trained to map its inputs back to the same inputs thus the output layer is identical to its input layer. The mapping (coding), bottleneck, and de-mapping (decoding) layers are the hidden layers. Typically auto-encoders are trained using the gradient descent method as in MLP however, it has been proposed by Bengio (2007) that gradient-based training of deep MLP networks gets stuck in the local minima or plateaus. In addition, it turns out that, although the performance function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. Thus several other algorithms have been applied to auto-encoder networks such as the scaled conjugate gradient backpropagation (SCG). In the conjugate gradient algorithms a search is performed along the conjugate directions avoiding the time-consuming line search at each iteration. This helps to produce generally faster convergence than steepest descent directions. The algorithm is based on conjugate directions and updates weight and bias values according to the scaled conjugate gradient method then backpropagation is used to calculate derivatives of performance with respect to the weight and bias variables. Detailed explanation of the algorithm can be found in (Moller, 1993). The memory requirements for this algorithm are relatively small in comparison to the other algorithms, so it is often a good choice for networks with a large number of weights. Moreover, it convergences faster than other algorithms and seems to perform well over a wide variety of problems including, approximation problems and pattern recognition problems Moller (1993) therefore; it was considered as the training function in the deep NN used in this research.

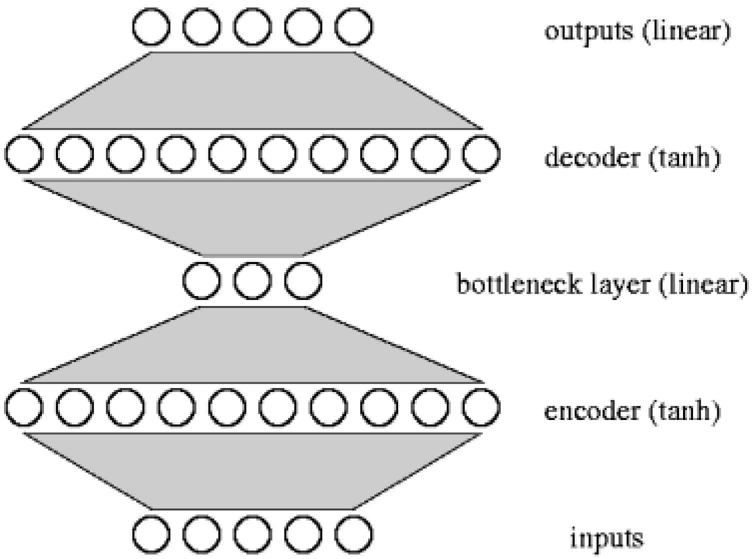


Figure [3.17]: Graph of a typical Auto-encoder neural network using encoder/decoder hidden layers. (Source: https://www.willamette.edu/~gorr/classes/cs449/nonsup.html)

3.9.5 Deep Belief Networks

Deep Belief Networks (DBN) have been introduced by Hinton and Salakhutdinov (2006) as stacked restricted Boltzmann machines (RBMs) that can be stacked and trained in a greedy manner to form so-called Deep Belief Networks (DBN) (Hinton,2006). DBNs are graphical models which learn to extract a deep hierarchical representation of the training data. The principle behind DBN is that good weight initialization plays an important role on the results therefore, Hinton (2006) and Bengio (2007) introduced a greedy layer-wise pre-training procedure, which is way of initializing better the parameters of DBN and after that can be applied to DBNs with RBMs as building blocks for each layer. Training of a DBN consists of two stages that allow learning feature hierarchies, as described by Hinton and Salakhutdinov (2006). In the first stage, generative unsupervised learning is performed layer-wise on RBMs. First, a RBM is trained on the data. Second, its hidden units are used as input to another RBM, which is trained on them. This process can be continued for multiple RBMs, as visualized in Figure [3.18]. In the second stage, fine-tuning using backpropagation is performed on the entire DBN to update the weights. Because of the pre-training, the weights have a good initialization, which allows backpropagation to quickly optimize the weights as described in Hinton and Salakhutdinov (2006) and (Sutskever et al., 2013).

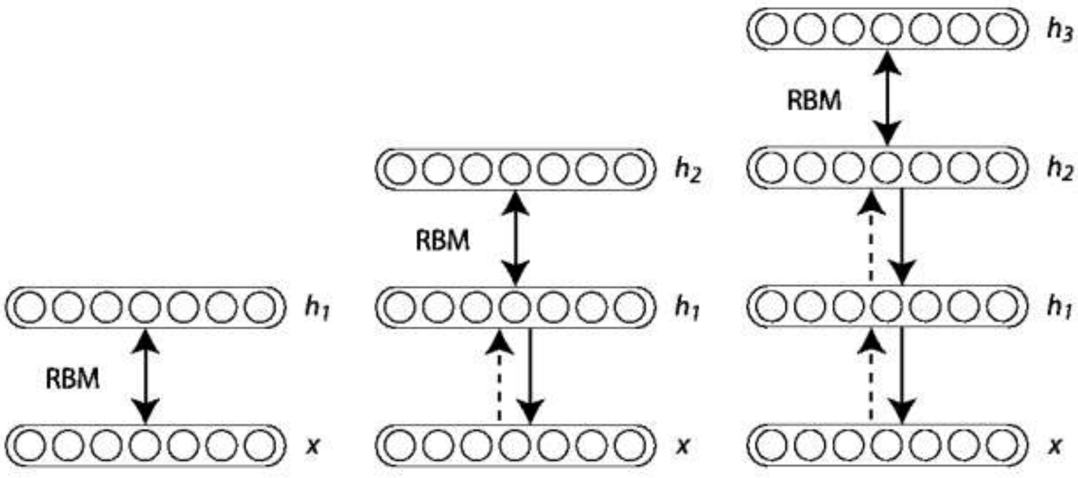


Figure [3.18]: Graph of DBNs

Layer-wise training of a DBN, composed by stacked RBMs. From the bottom, x is the input and *hk* are hidden layers. (1) The first layer is trained. (2) The second layer is trained using the first hidden layer as visible units. (3) The third layer is trained using the second hidden layer as visible units. The process can be continued for multiple RBMs then the resultant DBN is ready for the fine-tuning.

(Source: http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/DeepBeliefNetworks)

3.9.6 Relation between Models

As can be seen all described models tend to learn the feature representation of the data by compressing. The structure of MLPs compared to DBNs differs in the output layer and the direction of the connections between layers. In MLP networks the information flows from the input layer, through the hidden layer, up to the output layer. In a DBN the information flows both ways between the visible (input/output) layer and the hidden layer. The training process is also different in MLPs compared to DBNs. In MLP the training is based on generating randomly initialization weights where training in each DBN layer is trained independently and greedily first, and then fine-tuning using backpropagation is performed on the entire DBN. Because of the pre-training, the weights have a good initialization, which allows backpropagation to quickly optimize the weights. MLP with a single hidden layer form a shallow network that is able to approximate and model any function as proven in many literatures (Hornik (1991); Simon (1998); Samarasinghe (2006). However, multiple hidden layers form a deeper network architecture that can help to learn complex and complicated functions that can represent high-level abstractions and more effect representations Hinton (2006); and Bengio (2007). The deep network architecture has the potential to both improve the network generalization and to learn hierarchical representations of the input data and thus can better generalize to unseen data. Therefore, three different models will be explored as presented in chapter five to evaluate different neural network methods for learning the empirical task of memorizing the good design patterns and to investigate whether the use of Deep Belief Networks (DBN) adds any substantial gain over MLPs.

3.10 Summary

The chapter has presented the first successful software implementation of Tepper's 'Alignment Metric Engine'. In addition, the engine was augmented with the current state-of-the-art application for educational designs, the LDSE (Laurillard et al., 2011) tool to facilitate teaching practitioners to produce learning design patterns and objectively measure the degree and quality of those patterns. A schematic and a step-by-step illustration of the processes required to augment the alignment metric engine with the LDSE was presented. The limitation of the alignment metric engine was discussed in being theoretically based. Subsequently, this will be addressed by adapting it by incorporating an adaptive agent that is able to learn directly from 'good' design pattern examples to calibrate its system parameters and therefore enable it to make design decisions based not just on theory, but also those patterns that appear to 'work' in practice.

CHAPTER 4: Data and Methodology

This chapter introduces the research data methodology used for this research for collecting, analysing and integrating ‘real’ module design patterns into the alignment metric engine. The chapter starts by briefly describing the overall methodology deployed in collecting the required module designs and discussing the structure and format the module design data must adopt before it can be used by the system. The subsequent sections describe the methodology and the data collection process, which consists of a module design desk-based research study, in-depth observations and a checklist. The chapter concludes by explaining the statistical procedures used to analyse the collected data.

4.1 Systems’ Data

The alignment metric system in the previous chapter developed to measure how well aligned the educational components are when put together and guide teaching practitioners to use the ‘ideal’/aligned combination of learning outcomes and the other components based on the principles of constructive alignment. In order for the system to overcome the limitation of previous tools and to base its measure on theory *and* practice, the system needs to be extended to incorporate student satisfaction to enable the system to associate alignment value ranges with ‘good’ design practices. The system will then be able to adapt its parameters accordingly and make design decisions on the basis of theory and practice. Therefore, the system being developed needs to learn from learning designs of differing quality. In order to collate and pre-process the appropriate design pattern data, it was important to review the different learning design tools and their design patterns as seen in Chapter 2. Also it was important to identify the core components of the alignment metric and how these components inter-relate in a systemic way as seen in Chapter 2 and 3. Thus the system’s data need to focus on four components: learning outcomes, learning objectives, teaching and learning activities, and assessment tasks and how they correlate to student satisfaction. Specially that one of the research problem of this research is to examine the relationships between the main components of the teaching system and student satisfaction.

4.2 Research Methodology

For the purpose of this research, a desk-based research methodology was used for the data collection, which involves primary data research that seeks to obtain data directly from its original source. The design pattern data was extracted from a University Virtual Learning Environment (VLE). In-depth desk based observations covering 567 modules from the University's School of Science and Technology (spanning departments of Physics, Biology, Maths, Computing, and Chemistry) were conducted to collect the core educational design components (i.e. LOs, LObjs, TLAs and ATs) in structural design pattern format. The student satisfaction scores associated with the module design patterns were also captured. This approach was justified for the following reasons: firstly, there was a very limited amount of compatible LDSE design patterns publicly available (i.e. between 2013 and 2016 only a total of 122 design patterns became progressively available for research); secondly, the lack of interoperability or compatibility between design pattern structure of the different learning design tools available in the public domain. For example, LAMS design patterns were mainly focusing on sequencing the teaching activities (activity patterns) and therefore, not constructed in such a way that they captured other important components of the educational design such as learning outcomes and learning objectives. Finally, and most importantly, none of the current learning design tools are able to discriminate module design patterns based on their effectiveness in practice. Therefore, this research centres on the types of data available in the institution where courses and module design practices are underpinned by constructive alignment and therefore the module designs are generally structured in an outcomes-based way (although as will be explained, there is significant variation across modules requiring further review and development). The notion of 'well-designed' modules will be determined by good practice using student overall satisfaction scores as indicators. The level of overall student satisfaction according to the definition of overall student satisfaction used by the National Student Survey states that 'The National Student Survey (NSS) is a national survey, which has been conducted by Ipsos MORI annually since 2005. It gathers opinions from mostly final year undergraduates on the quality of their courses. Aimed at current students, the survey asks undergraduates to provide honest feedback on what it has been like to study their course at their institution' (NSS, 2013). Students rate their overall satisfaction based on level of agreement to a given question using a five-point Likert scale indicating the strength of their agreement with the statement (5 - Strongly Agree; 4 - Agree; 3 - Neither Agree nor Disagree; 2 - Disagree; 1 – Strongly Disagree). The institution applies a similar schema at module level for all modules within University by using the evaluation surveys system (EvaSys, 2013), which consists of 23 questions in six assessment categories including: Feedback on group-based teaching, Feedback on module teaching, Module organisation and resources, Overall satisfaction, School specific questions, and Student engagement. **Appendix [C]** contains an example of the student evaluation survey. The average score of each aspect is used to find the average satisfaction scores given to the evaluated module.

Reviewing and revising module designs according to measures of student satisfaction *and* constructive alignment may help module leaders to improve their module designs in a way that has tangible improvement in the student learning experience. The modules learning rooms were selected based on the 2012/13 and 2013/14 EvaSys scores of the associated module so that the data collected will be classified according to differing levels of student satisfaction. For this, the researcher chose a desk-based research methodology and designed checklist criteria to help to select the valid modules for reviewing and extracting the required data from. This approach however is expensive, time consuming and needs well-qualified, highly trained experts. Therefore, the researcher was required to have a clear research questions before the collecting data process begins to help to identify the scope and collect the appropriate data. The scoping process was driven by three underlying questions:

- What is needed?
- What is meaningful?
- What is the core data about the variable quality design patterns?

Answering these questions was based on mainly the main components of the alignment metric system and guided by those design patterns reviewed in Chapter two. Thus, the generated learning design patterns captured the module's key features as will be discussed later in this chapter. Having generating the required design patterns, this will form the next step where all data need to be pre-proceeded so that they can be transformed into a form that can be fed as input into the auto-encoder network. A set of auto-encoders with different configurations and hyper-parameters are considered for training the network models as detailed in Chapter 5 so that they act as perfect memories of good design patterns (those with high degrees of student satisfaction scores).

4.3 Ethical Consideration

For the purpose of the research, the data gathering process involved generating learning design patterns from selected module 'Learning Rooms' based on the 2012/13 and 2013/14 EvaSys scores. A successful Ethical Approval application was therefore made before embarking on this data collection stage. This ensured all affected Module Leaders within the target Schools were informed that their learning room will be accessed and reviewed for the data gathering process of this research and that this data will be anonymised during analysis and evaluation. All data were therefore anonymised in that module codes and references to specific staff were removed and the scope of the research would not involve collecting or analysing individual student data. Moreover, project-specific module identifier codes were created and assigned to ensure learning room data are anonymous.

4.4 Data Collection

Two methods were used during the data collection process. The first method involved using the checklist method. This method is known as an organizational method to improve effectiveness of a given task. It structures a person's observation or evaluation of a performance and helps to ensure consistency and completeness in carrying out a task. In addition to that, it provides a way of assessing that can help to limit the number of valid data (McNamara, 2008). A list of module codes and their EvaSys scores data was obtained from the school of science and technology. The list contains more than 400 module codes associated with their EvaSys scores data. In order to facilitate the process of collecting data from the given list, a module selection checklist with simple lists of criteria that can be marked as yes or no was created. This was created to help to select modules with clear module specification and contents that can be used to observe, investigate, and extract the required data from. Table [4.1] lists the checklist criteria. Applying the checklist criteria on total of 587 module designs, this excluded 9 modules as no module specifications were found, 4 modules as no clear module hand-book was provided, and 7 modules as no further information was listed on how assessment tasks are linked to the indented outcomes in the modules. This resulted in 567 out of 587 module designs were valid to observe more deeply and collect data from. Having identified the modules the next method was performed.

Table [4.1]: Module selection Checklist criteria

Checklist Criteria	Yes	No
Clear module specification available		
Further information on assessment task available in the specification		
Assessments are linked to the ILOs in the module specifications		
Clear teaching and learning activities available		
Module’s table of contents presented clearly		
Clear and descriptive module hand-out available		

The next step was to use the observation method to observe and review the selected modules to extract data and generate the design patterns. The Observation method is a way of gathering data by watching behaviour, events, or noting physical characteristics in their natural setting (Donald, 2005). Simply it can be described as the action or process of closely observing or monitoring something or someone. The desired data can be collected either directly by observing the event and taking notes or recording the event with electronic tools. This data collection method is most commonly used with qualitative data as described by many researchers (Beesey, Davie, and Savin, 1991; Saunders, Lewis and Thornhill, 2003; Mack and Woodsong, 2011). Patrick (2008) provides simple and sound common-sense advice on carrying out observations that was applied. After checking and identifying the modules using the checklist criteria, the observation method was conducted. This data collection method was used to observe the following: module specification, module hand-book, and the lectures, seminars, and labs in the module to extract and record (note down) the core qualitative data needed from learning outcomes, learning objectives, teaching and learning activities, as well as summative assessment tasks. The observation was conducted with the use of a Microsoft Excel spreadsheet to record and store the extracted data. The type and source of data collected is explained in Table [4.2] while simple flowchart diagrams are given in Figure [4.1] and Figure [4.2] clarifying the data collection procedure. The observation method has the advantage of producing a large amount of qualitative data and does not rely on people’s willingness or ability to provide information as in other data collection methods such as interviews and questionnaires. However; it is expensive, time consuming, and challengeable process thus some design considerations were made to facilitate the process and this is discussed in the next section.

Table [4.2]: Type of data collected during the observing and the place from where each data were extracted

Data to collect:	From where to collect
Module features	Module features are straightforward to collect from the module specification provided
Subject area	
Level	
Credit point	
Module's EvaSys scores	EvaSys scores of S1, S2, S3, and S4 are collected from the EvaSys excel spreadsheet provided
S1 EvaSys score of group-based teaching	
S2 EvaSys score of class-based teaching	
S3 EvaSys score of module organisation	
S4 EvaSys score of student satisfaction	
Module's core teaching and learning features	Some features are straightforward to collect from the main module specifications and module hand-out provided while others are challengeable and request deep investigation into module's content. The challenges encountered and the decisions made were addressed in Table [4.3].
Learning outcomes (verbs)	
Learning objectives (verbs)	
Teaching activities (TLA name)	
Assessment task (AT name)	

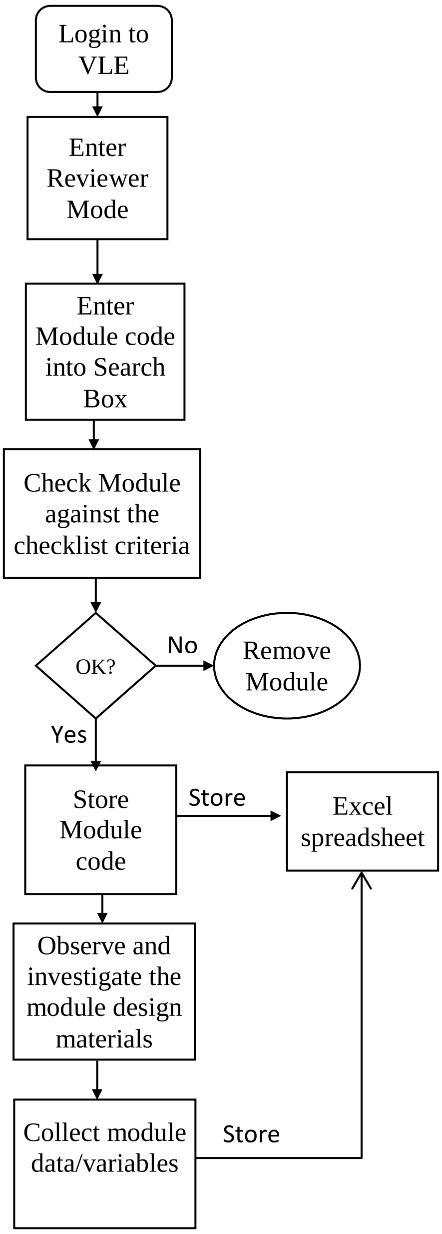


Figure [4.1]: Flowchart of the procedure of collecting the data

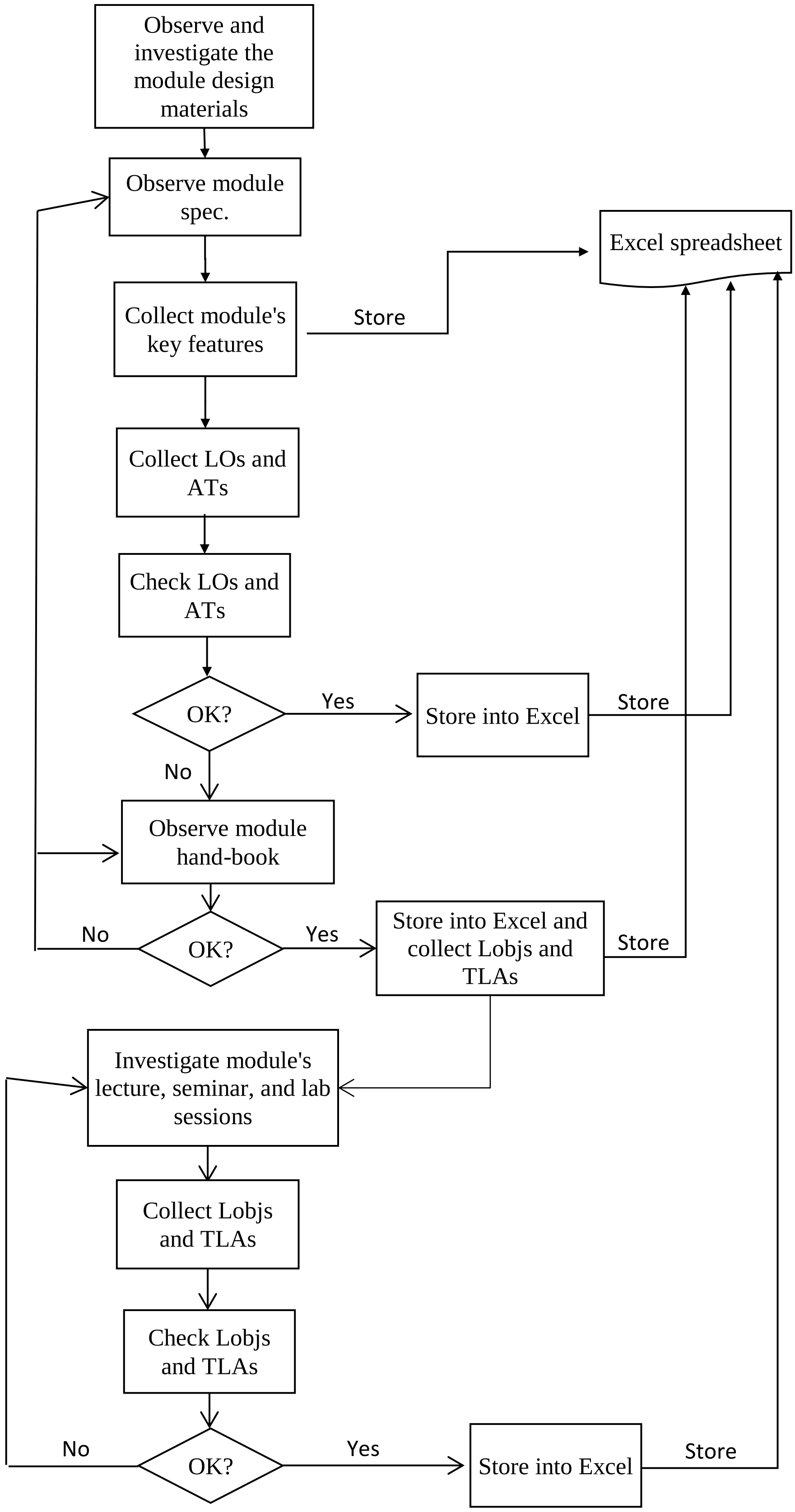


Figure [4.2]: Flowchart of the procedure of observing and investigating a module design to extract module features

4.5 Design Considerations

Data collection can be a technical, complex and expensive process, depending on the size, resources and needs. Yin (1994) mentioned that adopting research strategies for solution contribute to facilitate the process of the data collection and overcome the potential challenges. The data that needed to be collected from the module designs to generate the design patterns of the required structure was quite a challengeable task. Table [4.3] below summarises these challenges and strategies for solution to facilitate the data collection process.

Table [4.3]: Challenges in data collection and strategies for solution

Challenge/issue	Strategies for Solution
Learning objectives are missing, or not obvious like other components	• Investigate sessions • Search for TLAs within the sessions • Observe activities/statements of what teacher set during the session.
TLA and AT Terminology In SST and LDSE	• Map between the TLAs and ATs in SST and LDSE
Verbs, TLAs, and ATs sometime appear in more than one Bloom's level of the taxonomy.	• Index to the highest level in Bloom's's Taxonomy (Biggs, 2003; Tepper, 2006). (This is not a precise grouping and as Biggs noted, research into such groupings is so far incomplete and much work still needs to be done. However; it was used for the purpose of computing the alignment).
Number of learning outcomes, learning objectives, teaching and learning activates, and assessment tasks	• Take the minimum recommendation of LO by many literatures. • Collect five learning outcomes • Collect active, measurable, and assessable LO verb (Bingham, 1999; Fry et al., 2000; Biggs and Tang, 2007). • Avoid general and ambiguous verb such "understand", "know", "appreciate", etc. (Bingham,

	1999; Fry et al., 2000; Biggs and Tang, 2007); • Observe and collect two learning objectives for single learning outcome • Observe and collect two TLAs for single learning objective • Collect two ATs for a single learning outcome.

As seen from the table one challenge was that the different module designs available from NOW were of varying quality, ranging from those containing a complete set of data (for each desired component) to others that were missing the learning objectives data or perceiving learning objectives and learning outcomes as the same. This causes confusion and substantially increase the processing time of the data generation process. The learning objectives are defined as teacher-orientated and/or student-orientated statements that specify what activities the students need to perform to achieve the associated learning outcomes. Adopting the perspective of D'Andrea (1999); Tepper (2006); and Biggs and Tang (2007) that learning objectives are typically different from learning outcomes as objectives are more likely to be the input to the TLAs or the expressions of teachers' intentions to the TLAs as they determine the TLAs used, this helped to articulate the concept of learning objectives as more related to session activities. Thus the approach taken to overcome this issue was to investigate the sessions (lectures, seminars, labs) within the module to extract the learning objectives from, which are verb statements of the specific things which the teachers of the module intend to achieve during the given session.

Another challenge encountered was the TLAs and ATs terminology used in the School of Science and Technology (SST) at Nottingham Trent University and other module designs such as LDSE. Some TLAs and ATs in SST are more specific and diversified than the LDSE which are more general and packed under some categories. This involved initial mappings among the TLAs and ATs in SST and those in the LDSE to facilitate the collection and comparison of the learning activities and to be able to link in with the current implemented system. This is illustrated in **Appendix [D].**

A substantial challenge was to determine the number of learning outcomes, learning objectives, teaching and learning activities, and assessment tasks, and to keep the structure of the design patterns simple and consistent to allow for robust use and comparison. Dealing with module level data, as opposed to session level, means that there are many learning outcomes, learning objectives, and range of teaching, learning and assessment activities to collect. For the purpose of this research, we are seeking to generate pattern examples from the available module data to compute the alignment between the individual components and for the entire module. This requires the various tree structures to be generated which can then be mapped to vectors and matrices for competition. A subsequent requirement is therefore that these trees need to contain parent nodes that dominate a fixed number of child nodes i.e. need to be balanced via fixing the number of daughters (i.e. the valence) for each tree type. For this purposes, and to keep the structure of the pattern simple and consistent, and to facilitate the process of generating the pattern data, the number of learning outcomes, learning objectives, teaching and learning activates, and assessment tasks was empirically established based on the recommendations of some universities' quality handbook supplement for example, Nottingham Trent University, Leicester University, University of London. The quality handbooks recommend 12 – 16 learning outcomes for course design and between 5 – 8 learning outcomes for an optimal module design. Biggs and Tang (2007) also recommended that 5 intended learning outcomes are suitable for module/session design and the more intended learning outcomes; the more difficult it becomes to align teaching and learning activities and assessment tasks to each. For each module, a maximum of five learning outcomes was selected together with their associated elements of assessment task. It was noticeable that all SST modules used maximum of two elements of assessment with the possibility of breaking down each assessment element further into two components (e.g. a portfolio element may consist of a presentation and report). However; a single learning outcome in the module specifications was found to be assessed by a maximum of two components. There is no literature specifying the number of learning objectives or teaching activities within a session as this depends on the approaches of teaching and learning. On this basis, the restrictions of the data gathering process for each module are:

- Each learning outcome will dominate at most two learning objectives;
- Each learning objective will dominate at most two teaching and learning activities;
- Each learning outcome will be assessed by at least two assessment tasks.

Thus the extracted features are captured and stored in a simple structured design pattern in Microsoft Excel spreadsheet as shown in Figure [4.3]. This approach enabled a consistent focus to be applied to each module and one that would allow for useful design structures to be extracted. Each generated design pattern has the structure shown in Table [4.4]. For the purpose of clarification, an example of design pattern data generated from one of the learning design modules is illustrated in **Appendix [E]**. All the extracted data were transferred then from Excel spreadsheet into the metric's database to store the component and compute the alignment therefore, the developed data model form Chapter 3 was extended to included father entity related to the student satisfaction information as shown in Figure [4.4].

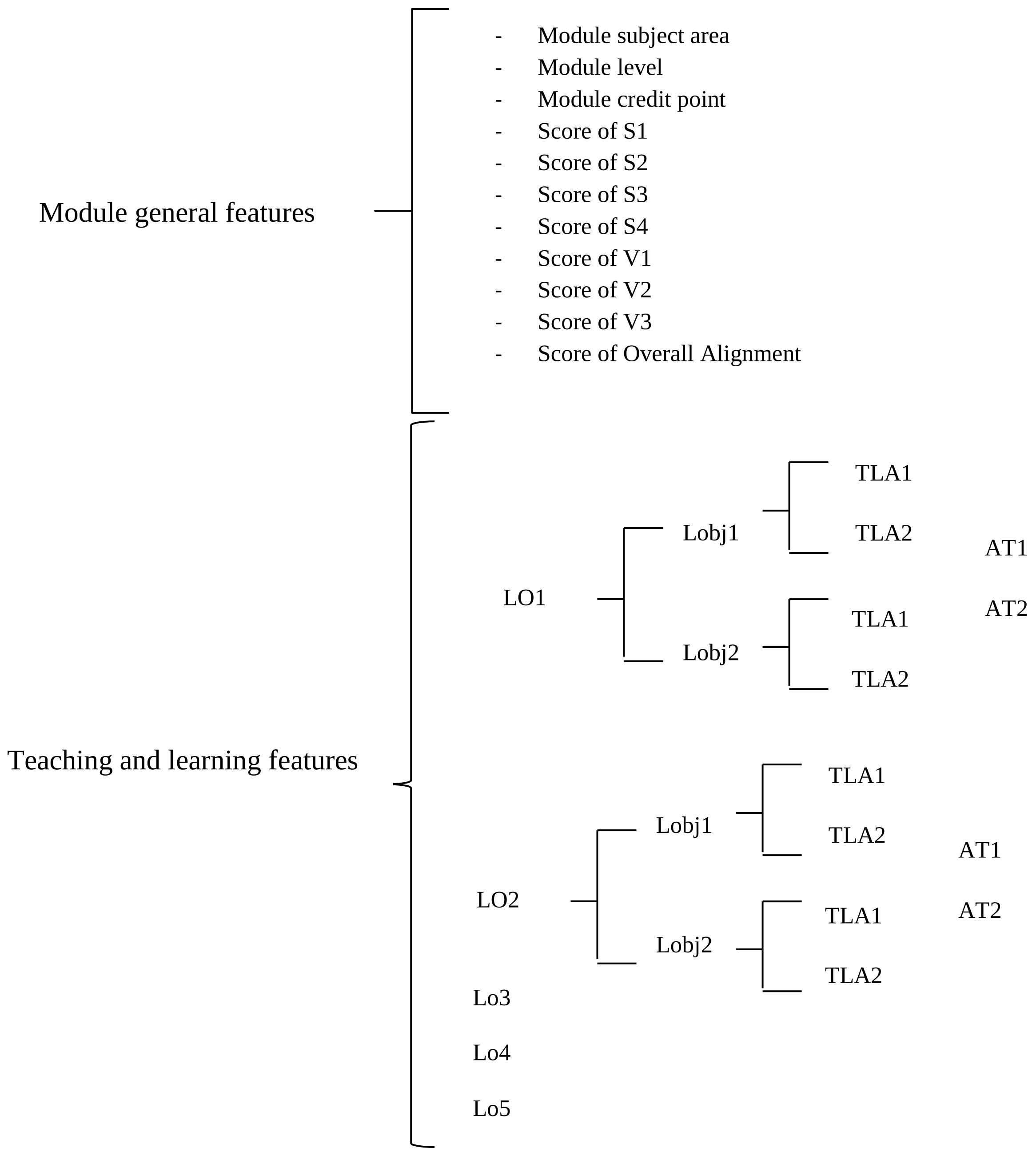


Figure [4.3]: Structure of Design Pattern Data in Excel

Table [4.4]: Design pattern data

Design pattern data
Module subject area
Module level
Module credit point
Score of S1
Score of S2
Score of S3
Score of S4
V1
V2
V3
OverallAlignment
Lo1- Learning outcome
Session Type
Lobj1- Learning objective
TLA1- Teaching and learning activity
TLA2- Teaching and learning activity
Lobj2- Learning objective
TLA1- Teaching and learning activity
TLA2- Teaching and learning activity
AT1- Assessment task
AT2- Assessment task
Lo2- Learning outcome
Session Type
Lobj1- Learning objective
TLA1- Teaching and learning activity
TLA2- Teaching and learning activity
Lobj2- Learning objective
TLA1- Teaching and learning activity
TLA2- Teaching and learning activity
AT1- Assessment task
AT2- Assessment task
Likewise for LO3, LO4, and LO5

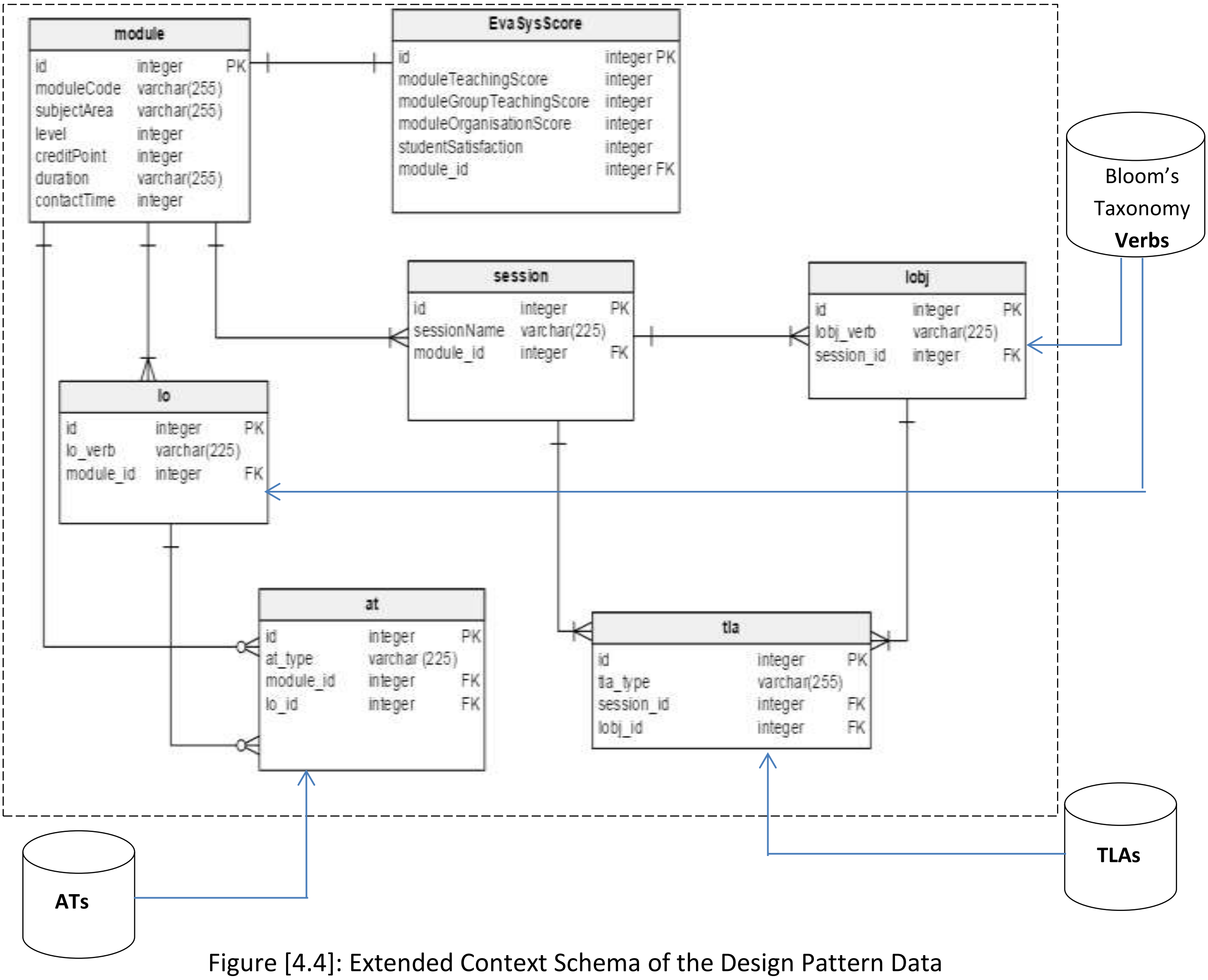


Figure [4.4]: Extended Context Schema of the Design Pattern Data

4.6 Understanding the Module Design Patterns Collected

As the module design pattern data has been collected, it is now necessary to understand the statistical nature of those data and divide it into two subsets, the first subset containing so called 'well-formed' module designs and the other, 'poorly-formed' module designs. The reason for this is that alignment metric engine will be augmented with a learning agent (in the form of an artificial neural network) to enable it to learn from 'well-formed' designs in a way that enables it to identify appropriate modifications to the 'poorly-formed' modules. Subsequently, for each module, the alignment scores were calculated between the design pattern components and then stored together with the student satisfaction scores. A total of 621 module design patterns were produced. To segment the data set into training and test samples, the design patterns were rank ordered according to the module's EvaSys score (where 5 refers to 'definitely agree' and thus excellent satisfaction and 1 to 'definitely disagree' and thus poor satisfaction). An EvaSys threshold score of 4 was selected and therefore the top 84% of the module designs were designated as training data and the remaining 16% as test data, as indicated below:

- Training dataset: 519 (84%) design patterns, all of which had student satisfaction scores of four or above and will be used to train the neural network about 'well-formed' module designs;
- Test set: 102 (16%) module design patterns, all of which had student satisfaction scores of 3.7 and below and used to evaluate/test the performance of the trained neural network. Note to understand the performance of the neural network and the design decisions it has made, the response of the network to each test pattern will be stored as the network is expected to produce a new pattern on the output layer in response to each test pattern. We will call the raw test set the 'TestSet (before)' sample and the new patterns formed by the neural network in response to the raw test set as the 'TestSet (after)' sample.

The remainder of this chapter discusses the statistical analysis and transformation performed on the module designs to aid learning. Data transformation of all patterns is based on the statistical properties of the training set. A frequency analysis was also performed to identify the most common parent-child relationships natural occurring within the module design patterns. More generally, the statistical properties of the Test Set (before) are computed so that they can be compared with those found in the resulting data set generated by the neural network (TestSet (after)) to understand the importance of the design decisions made by the system (as discussed in Chapter 6 and 7). Finally, it is important to state that we used the alignment tables presented in **Appendix [B]** to assign each TLA, AT to a Bloom's level based on Biggs (1999, 2003). Also we grouped similar verbs, activities and assessments that having the same Bloom's level together. For example, all TLAs that were assigned to Bloom's level 2 ,according to the TLA alignment table, such as 'lecture', 'tutorial', 'online presentation' will be grouped under TLA(2), activities such as 'seminars' and 'class discussion', which were assigned to Bloom's level 3, will be grouped together under the TLA(3). The same procedure was applied for the other activities and assessment tasks and the full list is provided in **Appendix [F]**. The following are variables that form the core part of the study of the raw data.

- V1: the learning outcome alignment score
- V2: the teaching and learning activities alignment score
- V3: the assessment tasks alignment score
- V: the overall module alignment score. This is obtained by taking the average of V1, V2 and V3.
- S: the overall student satisfaction score

4.6.1 Statistical Analysis of the Training Set

The training data set consists of 519 module design patterns where each pattern has a student satisfaction scores of four or above. The data were analysed in terms of the V1, V2, V3, V and S scores and the statistical descriptive analysis of the data is given in Table [4.8] which is illustrated at the end of this section. The design patterns data were also subject to frequency analysis to identify and describe the module characteristics that has significant impact on overall student satisfaction. The following graph illustrates the overall frequency of Bloom's level for each of LOs, LObjs, TLAs, and ATs found in the training dataset.

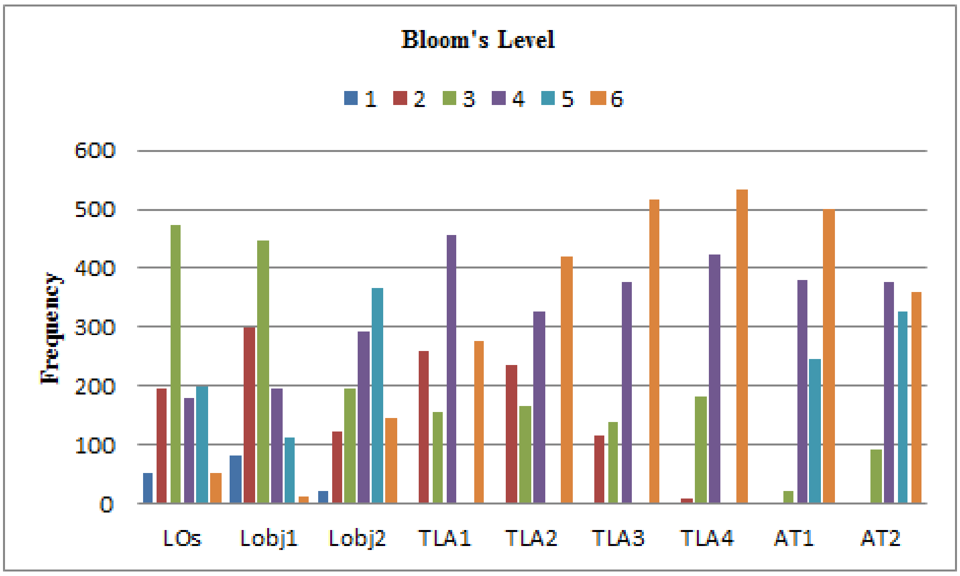


Figure [4.5]: Frequency graph showing the most frequent Bloom's learning level covered in the training dataset

A frequency table of each parent-child relationships for each of V1, V2, and V3 tree was constructed as shown in the tables below. Table [4.5] shows the most common learning outcome/learning objective parent-child relationships found in the training dataset (i.e. in the well-formed module designs according to student satisfaction). For each learning outcome found in the module designs, the most frequent corresponding learning objectives are indicated.

Table [4.5]: V1 Frequency Relationships Table (Training dataset)

Lo/Lobjs	Lo(1)	Lo(2)	Lo(3)	Lo(4)	Lo(5)	Lo(6)
Lobj(1)	12 (8.9%)	29 (7.3%)	34 (3.6%)	11 (3.0%)	22 (5.5%)	0 (0%)
Lobj(2)	53 (52.4%)	175 (44.2%)	131(13.8%)	44 (12.3%)	18 (4.5%)	3 (3.0%)
Lobj(3)	14 (13.8%)	79 (19.9%)	359 (38.0%)	80 (22.5%)	110 (27.9%)	9 (9.2%)
Lobj(4)	20 (19.8%)	84 (21.4%)	169 (17.9%)	74 (20.6%)	100 (25.3%)	34 (35.0%)
Lobj(5)	4 (3.9%)	13 (3.2%)	219 (23.1%)	98 (27.5%)	123 (31.2%)	16 (16.4%)
Lobj(6)	1 (0.9%)	16 (4.0%)	32 (3.3%)	49 (13.7%)	21 (5.3%)	35 (36.0%)

For each of the learning objective in the module designs, the frequent teaching activities are the following:

Table [4.6]: V2 Frequency Relationships Table (Training dataset)

Lobj/TLAs	Lobj(1)	Lobj(2)	Lobj(3)	Lobj(4)	Lobj(5)	Lobj(6)
TLA(1)	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**
TLA(2)	**204(52.3%)**	**572 (38.1%)**	**150 (5.1%)**	**216 (10.2%)**	**238(10.2%)**	**73 (10.6%)**
TLA(3)	**97 (24.8%)**	**380 (25.3%)**	**370 (12.5%)**	**108 (5.1%)**	**32 (1.6%)**	**305(44.3%)**
TLA(4)	**67 (17.1%)**	**301 (20.0%)**	**840(28.5%)**	**767 (36.2%)**	**895(36.6%)**	**7 (1.0%)**
TLA(5)	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**
TLA(6)	**22 (5.6%)**	**247 (16.4%)**	**1578(53.7%)**	**1023(48.3%)**	**1157(49.6%)**	**303(44.0%)**

For each of the learning outcome in the module designs, the frequent assessment tasks are the following:

Table [4.7]: V3 Frequency Relationships Table (Training dataset)

Lo/ATs	Lo(1)	Lo(2)	Lo(3)	Lo(4)	Lo(5)	Lo(6)
AT(1)	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**
AT(2)	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**
AT(3)	**0 (0%)**	**44 (6.0%)**	**80 (3.7%)**	**0 (0%)**	**35 (3.6%)**	**46 (20.0%)**
AT(4)	**39(22.1%)**	**178(24.5%)**	**865(41.0%)**	**244(32.1%)**	**436(45.8%)**	**41 (17.8%)**
AT(5)	**37(21.1%)**	**44(6.0%)**	**628(29.7%)**	**236(31.0%)**	**44(4.6%)**	**39 (16.9%)**
AT(6)	**100(56.8%)**	**460(63.3%)**	**535(25.3%)**	**279(36.7%)**	**436 (45.8%)**	**104(45.2%)**

Resultantly, the mean and standard deviation (SD) are evaluated for V1, V2, V3, and S. in addition, the range which is the difference between the minimum and maximum value, is also calculated for these variables as summarised in Table [4.8] below. The mean and SD are given correct to 2 decimal places while the minimum and maximum values and the range are estimated correct to 1 decimal place.

Table [4.8]: Training Dataset Statistical Summary

	V1	V2	V3	V	S
Mean	7.91	9.26	10.45	9.21	4.52
SD	3.21	2.39	2.74	1.62	0.28
Min	4.8	4.3	4.3	4.7	4.0
Max	19.8	16.3	17.6	13.5	5.0
Range	15.0	12.0	13.3	8.8	1.0

The frequency of usage of each LOs, LObjs, TLAs, and ATs as seen from the graph in Figure [4.5] allows making the conclusions about their importance and significance (occurrence) in the design patterns. It can be seen that the most considered Bloom's cognitive in formulating learning outcomes is Apply with more than 450 design patterns (41%) occurring with verbs under Bloom's level 3. This is followed by Create with 17.4% of the outcomes were in Bloom's level 5 and 16.9% were in Bloom's level 2 (i.e. Understand), while design patterns with learning outcomes in the Analysis level were occurring with 15.5%. On the other hand, design patterns with learning outcomes associated with Bloom's level 1 and 6 were less frequent as there were no more than 4.5% of design patterns with verbs in learning outcomes of Knowledge and Evaluation. This means that for the majority of learning outcomes of these modules, emphasis was more on the intermediate cognitive level in formulating the learning outcomes as the highest proportion of frequency was at Bloom's level 3. Following that it can be seen that the most frequency of Bloom's levels for formulating the learning objectives in general were between level 2, 3, 4, and 5 with the first learning objective was usually utilizing the same Bloom's level of the most frequent learning outcome (38.9%) while Bloom's levels in the second learning objective have been mixed with more frequency for higher cognitive skills between 32% for Bloom's level 5 (Create) and 26% for Bloom's level 4 (Analyse) that is students being expected to create and analyse the artefacts.

In terms of the teaching and learning process, it can be seen that there are wide range of activities assigned to different level of Bloom's with a great emphasis (i.e. more than 38%) on more high level activities associated with level 6, which contains all forms of group-based activities. Also it was found that active-based learning and collaborative learning were the most type of learning constructed during the given lectures. Finally, the most frequent assessment tasks used were the assessment types which assigned to Bloom's level 4 and 6 and frequently taking the form of individual practical assessment and reports, marked assignments, group projects, and 2 hours unseen examinations.

The frequency table of each parent-child relationships constructed to determine the most frequent or common parent-child relationships found in the training set, shows that in V1 relationships generally there is a good balance between each of the learning outcomes and learning objectives. It shows that learning outcomes formulated with more emphasis on the ability of students to remember knowledge and understand (Bloom's level 1 and level 2) are associated with the verbs used in formulating the learning objectives by containing one or more of the verbs under "Understand and Analyse" (level 2 and level 4). This shows that the associated learning objectives utilizing the same or higher levels than the learning outcomes. On the other side, learning outcomes constructed on the ability of students to apply, analyse, and create (level 3, level 4, and level 5 respectively) are commonly associated with learning objectives contain one or more of the verbs under "Apply and Create" (level 3 and level 5). And finally, learning outcomes with the highest cognitive abilities (Bloom's level 6) seems to be more associated with learning objectives contain one or more of the verbs under "Analyse and Evaluate" (level 4 and level 6).

The V2 relationship, which illustrate the relation between the learning objectives (LObjs) and the dominated teaching and learning activities (TLAs) used, shows that the most common types of TLAs used when the verb of the learning objective being in Bloom's level 1 are lectures, which assigned to Bloom's level 2, and variety of seminars and class discussions (Bloom's level 3). A similar trend was observed for the verbs of the learning objective being in Bloom's level 2. This is considered appropriate because verbs related to these levels refer to declarative knowledge that helps students to learn and know about certain topics or facts and so, the appropriate activities to facilitate achieving this are teaching activities like lectures, seminars, and discussion as mention in (Biggs and Tang, 2007). In learning objectives being formulated at higher level skills like Bloom's level 3, 4, 5, and 6, the most frequent TLAs found are those which assigned to higher levels in Bloom's including more peer-controlled and a combination of individual and group work activities during the teaching session with high percentages associated with group-based activities (TLA (6)) as they were the most frequent activities. This is achieving the principle of collaborative learning that helps students to work together to achieve high form of learning. It is well documented in many literatures that collaborative learning is one of the most attracted and valued approaches in learning as it combines social learning with experiential learning or inquiry-based learning in the sense that students work together in pairs or small groups to construct common meaning and knowledge and to produce and demonstrate the outcome of their learning (Bandura, 1985; Vygotsky, 1978; Roschelle, 1992; Brett, 2005; Machemer and Crawford, 2007; Cavanagh, 2011, and others). Therefore, it is understandable that module designs containing high proportions of collaborative methodologies hold high student satisfaction as they help better understand the learning which leads to a positive effect. In addition, Vygotsky (1978) had mentioned the capability of students to achieve higher intellectual levels when asked to work collaboratively than individually.

Finally, the relationship between the learning outcomes and assessment is one of the most important factors in achieving well aligned module design. Biggs and other education scholars emphasised the importance of designing assessments and ensuring that they match the learning outcomes of the module so that student can be assessed using the right level. Table [4.5] illustrates the most frequent assessment tasks used for each level of the learning outcomes in the training dataset. In general it can be seen that the most frequent Bloom's levels used to assess the learning outcomes were Bloom's level 4 and 6. Assessment tasks assigned to Bloom's level 4 are types such as in-class tests, practicals, and reports. And assessment tasks assigned to Bloom's level 6 are types such as exams, essays, and projects. In analysing the assessment types used for each learning outcomes, the table shows that the most frequent type of assessments in assessing learning outcomes related to the knowledge and understanding (Bloom's level 1 and 2) are exams (Bloom's 6), with 560 learning outcomes found to be on this pattern. In addition to exams, these learning outcomes were assessed with short answer exams which were taken as in-class tests (Bloom's 4). For learning outcomes being at the application level, where the emphasis is more on the ability of the student to apply the learnt knowledge to solve problems, 41% of the outcomes were assessed mostly by individual practical activities and reports while 29% of the outcomes of the same level were also assessed by more individual depth assignments. On the other hand, the most frequent combination of assessment tasks in assessing the learning outcomes in the analyses and synthesis levels were Bloom's level 6 in the form of group assignments and/or projects and Bloom's level 4 in the form of individual practical reports. Finally, for learning outcomes at the evaluate level, the most frequent assessment strategy involved was examination as 45% of the learning outcomes being in this level were assessed by unseen examination. In addition to that, there were other assessment types used beside the examinations to assess the outcomes such as presentations, individual reports, and project reports with the frequency proportion of these types given as 20%, 17%, and 16% respectively.

4.6.2 Statistical Analysis of the Test Set (Before)

The same analysis principles were applied for the testing dataset which consist of 102 data patterns that were generated from the 'poorly-formed' module designs. The data were analysed in terms of the alignment scores and satisfaction scores. The statistical descriptive analysis of the data is given in Table [4.12] where the overall frequency analysis of LOs, LObjs, TLAs, and ATs is given in Figure [4.6].

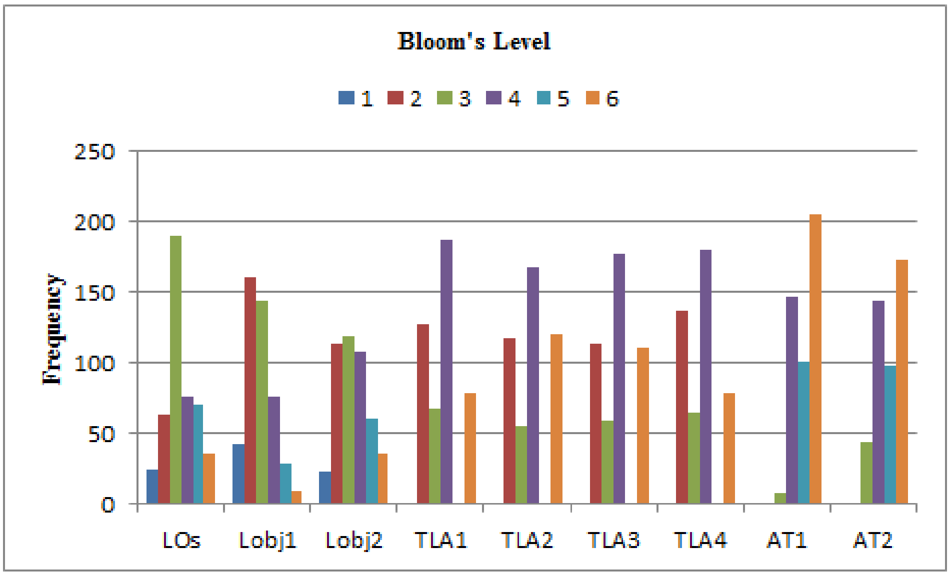


Figure [4.6]: Frequency graph showing the most frequent Bloom's learning level covered in the testing dataset

Again once the overall modules taxonomy was analysed for the testing dataset, a frequency table of each parent-child relationships in each of V1, V2, and V3 tree is constructed. This to determine what the overall pattern or the most common parent-child relationships found in the testing dataset. For each of the learning outcome in the module designs, the frequent learning objectives are the following:

Table [4.9]: V1 Frequency Relationships Table (Test before)

Lo/Lobjs	Lo(1)	Lo(2)	Lo(3)	Lo(4)	Lo(5)	Lo(6)
Lobj(1)	2 (3.1%)	18 (13.0%)	23 (5.2%)	7 (4.2%)	6 (4.1%)	0 (0%)
Lobj(2)	33(51.5%)	57 (41.3%)	72 (16.5%)	33 (19.8%)	12 (8.3%)	6 (8.3%)
Lobj(3)	16 (25.4%)	24 (17.3%)	173(39.8%)	39 (23.4%)	58 (40.5%)	20 (27.7%)
Lobj(4)	13 (20%)	31 (22.4%)	72 (16.5%)	47 (28.3%)	34 (23.7%)	28 (38.8%)
Lobj(5)	0 (0%)	5 (3.6%)	80 (18.4%)	25 (15.0%)	32 (22.3%)	12 (16.6%)
Lobj(6)	0 (0%)	3 (2.1%)	41 (3.2%)	15 (9.0%)	1 (0.6%)	6 (8.3%)

For each of the learning objective in the module designs, the frequent teaching activities are the following:

Table [4.10]: V2 Frequency Relationships Table (Test before)

Lobj/TLAs	Lobj(1)	Lobj(2)	Lobj(3)	Lobj(4)	Lobj(5)	Lobj(6)
TLA(1)	0(0%)	0(0%)	0(0%)	0(0%)	0(0%)	0(0%)
TLA(2)	52(45.6%)	155(36.3%)	353(53.0%)	240(53.6%)	27(8.7%)	4(5.1%)
TLA(3)	28(24.5%)	81 (19.0%)	108(16.2%)	60(13.4%)	60(19.3%)	0(0%)
TLA(4)	26(22.8%)	128(30.0%)	152(22.8%)	113(25.28%)	169(54.5)	41(52.5%)
TLA(5)	0(0%)	0(0%)	0(0%)	0(0%)	0(0%)	0(0%)
TLA(6)	8(7.0%)	62(14.5%)	53(7.9%)	34(7.6%)	54(17.4%)	33(42.3%)

For each of the learning outcome in the module designs, the frequent assessment tasks are the following:

Table [4.11]: V3 Frequency Relationships Table (Test before)

Lo/ATs	Lo(1)	Lo(2)	Lo(3)	Lo(4)	Lo(5)	Lo(6)
AT(1)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
AT(2)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
AT(3)	3 (4.5%)	9 (6.5%)	37 (8.4%)	8 (4.8%)	8 (5.6%)	3 (4.1%)
AT(4)	28(42.4%)	40 (28.9%)	183 (41.7%)	53 (31.9%)	55 (38.7%)	14 (19.4)
AT(5)	10(15.1%)	29 (21.0%)	88 (20.9%)	44 (26.5%)	34 (23.9%)	18 (25%)
AT(6)	25(37.8%)	60 (43.4%)	130 (29.6%)	61 (36.7)	45 (31.6%)	37 (51.3%)

The statistical figures for each of the V1, V2, and V3 scores and student satisfaction scores in the testing dataset are given as follows again the mean and SD are given correct to 2 decimal places while the minimum and maximum values and the range are estimated correct to 1 decimal place.

Table [4.12]: Test Dataset Statistical Summary

	V1	V2	V3	V	S
Mean	9.60	7.81	10.15	9.19	3.36
SD	5.24	2.39	3.14	2.75	0.31
Min	3.0	3.7	4.0	4.4	1.8
Max	21.2	16.4	16.7	14.3	3.7
Range	18.2	12.7	12.7	9.8	1.9

The frequency of usage of each LOs, LObjs, TLAs, and ATs as seen from Figure [4.6] illustrates that again the most considered Bloom's cognitive in formulating learning outcomes is Apply with 41% of the outcomes contain different verbs under the Bloom's level 3. This is followed by Analyse, Create, and Understand with very close percentages of 16.5%, 15.2%, and 13.7% respectively. Again design patterns with learning outcomes associated with

Bloom's level 1 and 6 were less frequent, however; the percentage of learning outcomes at Bloom's 6 in this set is slightly more than what found in the training set. The majority of the learning objectives were constructed around the Understand, Application, and Analysis levels of Bloom's taxonomy. In terms of the teaching and learning process, again the teaching was mainly designed through lecture sessions, seminars, and practical sessions where the most frequent type of teaching activities beside lectures were those TLAs assigned to Bloom's level 4 with high proportions of frequencies for using the individual-based activities as main TLAs. It is clear from the graph that TLAs with Bloom's level 6, which consist of group-based activities, were less frequent with an average occurrence of only 20% of the overall TLAs used. This explains that modules' leaders of these modules were focusing on individual activities whether it is practical or resource-based more than engaging the student to work together. Last but not least, the most frequent assessment tasks in this dataset seems to be slightly the same as the training dataset in utilizing Bloom's level 4, 5, and 6 in assessing the learning outcomes and taking the form of individual/group assignments, practicals, and unseen examinations.

In terms of the frequency table of each parent-child relationships for V1, V2, and V3, this shows that there is some mismatch tacking place. For example, in V1 relationship (Table [4.9]), the learning outcomes which assess students ability to create, and evaluate were frequently associated with learning objectives in the lower Bloom's cognitive abilities than that defined in the learning outcomes. According to principles of constructive alignment this can limit the achievement of the outcome because low level cognitive learning objectives will not dominates the right type of TLAs that can help students to create and evaluate even if they achieved the defined objectives. The reason for this may be due to the confusion in the understanding of the concept of the relationship between learning outcomes and learning objectives as it was explained before in that most teaching practitioners do not differentiate between the two with only a number of researchers such as D'Andera's (1999) making it clear that learning objectives are the input to the TLA process which requires the teaching practitioners to make more sensible choices and design wide range of activities to get students to do what the learning outcomes nominate. However, considering wide range of activities is not enough, the types of TLAs also play an important role in education. For example, the frequency table of V2 relationship shows that the most frequent type of TLAs associated with learning objectives related to Bloom's level 1(Knowledge) were lectures and discussions, Bloom's level 2 (Understanding) were lectures and 30% individual activities, Bloom's level 3 and 4 (Application and Analysis) were also mainly lectures (53%) and more than 20% individual activities, and finally Bloom's level 5 and 6 (Synthesis and Evaluate) were also associated with high propositions of individual activities and other small percentages distributed among lectures, seminars, and little of group activities. in this dataset, lectures were the most frequent TLA, which is strong in achieving lower-cognitive objectives but not suitable for the higher objectives.

The frequency table of V3 relationship revealed that the most frequent assessment tasks in this dataset seems to be similar to the training data. It can be seen that one of the reasons that these modules are associated with low student satisfaction may be the lack of social and collaborative activity as seen in the V2 table with less frequencies for the group activities, which is opposite to the training set that comprises more lager proportions on the group activities than on the individual ones. Many of scientists and pedagogical educators have asserted that both individual and group activities play an important role in the learning, however; social and collaborative activities remain the most influential.

The results of the analysis on the cognitive levels associated with learning outcome, learning objectives, teaching activities, and assessment tasks, show that there are sufficiently many examples of good practices in aligning the components to required level based on the principle of constructive alignment as seen in the training sets. However, there is also a number of design patterns that suffer from weakness in the teaching methods which need to consider other teaching methods in order to attract student satisfaction. The next section will highlight the distinct differences between the two sets where the following section examines and discusses the relationships between each of the V relationship and student satisfaction in design patterns generated from module designs with high student satisfaction scores. Each relation is examined separately where the Pearson correlation coefficient and R-squared are calculated.

4.6.3 Brief Comparison between the Training and Test Sets

The aim of conducting the above frequency analysis was to identify the design features in the dataset and highlight the most distinct differences between them. As demonstrated by the above frequency graphs and tables, there was a nice colour variation in how the learning outcomes are formulated with more focus on the application level in both sets. However, there was a difference in associating the other components together in relation to the learning outcomes. Table [4.13] below briefly highlights these distinct differences between the training and test dataset which resulted from the frequency analyses conducted for the V1-LO/Lobj, V2-Lobj/TLA, and V3-LO/AT relationships.

Table [4.13]: Comparison table between the Training and Test Sets

	Training Set	**Test Set**
LO/Lobj	Objectives levels are more distributed among intermediate to high abilities: The level of learning associated with analysis (4) and synthesis (5) are most frequently associated with objectives at application (3), Analysis (4) and synthesis (5). The level of learning associated with evaluation (6) are most frequently associated with objectives at analysis (4) and evaluation (6).	Objectives levels are more distributed among low to intermediate abilities: The level of learning associated with analysis (4) and synthesis (5) are most frequently associated with objectives at application (3) and analysis (4). The level of learning associated with evaluation (6) are most frequently associated with objectives at application (3) and analysis (4).
Lobj/TLA	TLAs in the form of group–based activities are always used in relation to the application, analysis, synthesis, and evaluation.	TLAs in the form of individual–based activities are frequently used in relation the application, analysis, and synthesis.
LO/AT	Assessment tasks assigned to Bloom’s level 3 and 6 are frequently used in relation to learning outcome at level 6. Presentation and examinations mostly used to assess learning outcomes associated with the evaluation level.	Assessment tasks assigned to Bloom’s level 5 and 6 are frequently used in relation to learning outcome at level 6. Individual/group assignments and examinations mostly used to assess learning outcomes associated with the evaluation level.

Before testing the relationship between the module alignment and student satisfaction, a two-sample t-test is performed for comparing the means of V1, V2, and V3 of the training dataset and testing dataset. This is done to determine whether there is a statistically significant deference between the means in the two groups or not thus the hypothesis of interest can be expressed as:

H0: There is no significant difference between the means found in the training dataset and testing dataset

In order to test the above hypothesis, the two-sample t-test used and the calculation results are obtained in Table [4.14] where figures are given correct to 3 decimal places. The results show that for V1 the t-statistic is equal to 3.142 and the p-value is very low, therefore, we reject the null hypothesis for V1 and conclude that there is strong evidence of a mean difference between the two sets. For V2, the t-statistic is equal to 5.553 and the p-value again is very low. Since the p-value is very low, we reject the null hypothesis for V2 as well and conclude that there is strong evidence of a mean difference between the two sets. However, for V3 the null hypothesis cannot be rejected because the p-value seems to be greater than the significant level 0.05 which shows no significant evidence of a mean difference between the two sets in the case of V3.

Table [4.14]: t-Test: two-sample assuming unequal variances

	V1	*V2*	V3
V_Mean_1	7.914	9.262	10.454
V_Mean_2	9.606	7.814	10.154
t-Stat	3.142	5.553	0.895
Sig.	< 0.000	< 0.000	> 0.371
Df	117	145	134

* Significant at the $p < 0.05$ level.
V_Mean_1 the V mean of training dataset (i.e. module design with high student satisfaction)
V_Mean_2 the V mean of testing dataset (i.e. module design with low student satisfaction)

4.6.4 Relationship between Module Alignment and Student Satisfaction

The next analysis is conducted to look for the correlation between the different alignment scores and student satisfaction scores in the module designs to investigate how well the scores are related, thus the hypothesis for this can be stated as:

H0: There is no correlation between the module alignment scores and student satisfaction scores in the module designs.

In order to investigate this hypothesis, first a scatter plot of the overall module alignment scores and student satisfaction scores was given in Figure [4.7] to visualize the underlying trend in the relationship. We can perceive from the scatter plot that there is an underlying assumption of a relationship between the overall module alignment and student satisfaction.

After that the different relationships were examined separately using the Pearson correlation coefficient. The correlation results in Table [4.15] suggest the existence of correlation between the different variable V1, V2, and V3 and satisfaction with different degrees. As can be seen that student satisfaction scores with V2 alignment scores (r = 0.575) is relatively highly correlated in comparison with V1 and V3 alignment scores. The table also supports the above assumption with a value of (r = 0.743), which suggests that an increase in the overall module alignment results in a corresponding increase in the satisfaction and vice versa. Therefore, we reject the above hypothesis and conclude that there is a positive correlation between alignment and satisfaction.

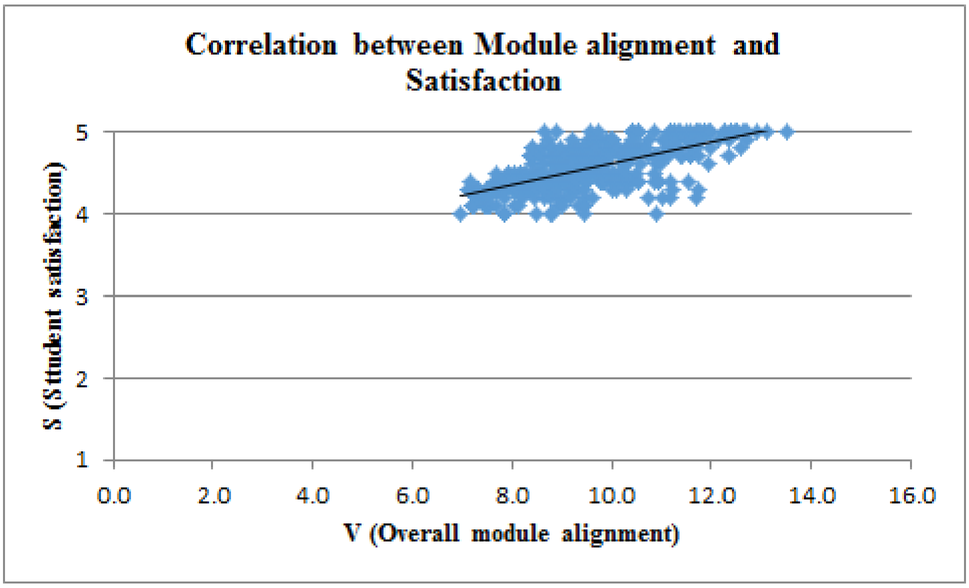


Figure [4.7]: Correlation plot between overall module alignment and student satisfaction

Table [4.15]: Correlation analysis between V1, V2, and V3 alignment scores and student satisfaction scores

	Pearson Correlation	*P-value*	*R-squared*
V1 and S	0.403	< 0.05	0.162
V2 and S	0.575	< 0.05	0.331
V3 and S	0.325	< 0.05	0.105
AVG_V and S	0.743	< 0.05	0.553

* Correlation is significant at the 0.05 level.

4.6.5 Determining Acceptable and Meaningful Range of Alignment Scores

The current alignment metric (Tepper, 2006) provides a quantitative measure of alignment between the individual components V1, V2, and V3 and for the entire module, which is theoretically based measure. In order to investigate the usefulness of the metric and to identify more realistic values for V1, V2, and V3, we use other indicators such as good design practices based on high student satisfaction to bridge between theory and practice and enable the metric to base its measure not only on theory but also on good and effective design practices. Therefore, the collected module data were arranged by their average EvaSys scores with the top 75% module designs being used for producing a meaningful alignment system where acceptable ranges are based on high level of student satisfaction, rather than theory alone. The 519 design patterns generated from the top 84% module designs were fed into the Alignment Metric where the relationships between the main components (i.e. LOs, LObjs, TLAs, and ATs) and Bloom's taxonomy were established and the alignment was computed. The statistical properties of alignment for V1, V2, and V3 have been calculated as shown in Table [4.8] in the previous section. The z-scores values are used to determine the range of the acceptable alignment scores that are subsequently used to evaluate the test data set before and after. The z-score for a given datum x is calculated using equation (3) below and the preliminary acceptable raw alignment values were identified as tabulated in Table [4.16]. It was determined to set the acceptable ranges for the alignment figures for V1, V2, and V3 respectively to be within the average of plus or minus one z-score range (69% training data) as more than 30% of the test patterns in each V1, V2, and V3 found to be outside this range.

$$z = \frac{x - mean}{std} \quad \text{..} \quad (3)$$

Table [4.16]: Acceptable Ranges of Alignment for V1, V2, V3, and AVG_V

	Number (X)	**% (X)**	**Min (X)**	**Max (X)**	**Range (X)**	**Z-scores**
V1	434	83.6	4.8	11.1	6.3	-0.96 – 0.99
V2	420	80.1	6.9	11.5	4.6	-0.98 – 0.93
V3	431	83.0	7.8	13.1	5.3	-0.96 – 0.96
AVG_V	432	83.2	7.6	10.7	3.1	-0.99 – 0.94

4.7 Summary

The purpose of this chapter was to describe the data research methodology of this research, explain the module design data selection, describe the procedure used in designing the tool and collecting the data, and provide an explanation of the statistical procedures used to analyse the collated data. Moreover, the training dataset, which represents the good and effective module design practices, was used to identify meaningful alignment value ranges for the three main relations (V1, V2, and V3) for Tepper's metric (Tepper, 2006). Applying the metric to the module design patterns in the training set resulted in the alignment value ranges shown in Table [4.16]. Therefore it is expected that if module designs stay within these ranges then the modules will be well-formed and constructively aligned in a way that will potentially yield positive student satisfaction. The next chapter presents the neural network data pre-processing along with the neural network training experiments for learning the features of good design patterns.

CHAPTER 5: EDIT an Educational Design Intelligence Tool for Supporting Design Decisions

5.1 Data Pre-Processing

In order to prepare the collected data for the artificial neural network, the data need to be pre-processed so that the original raw data is transformed into numerical input vector or feature vectors ready to be fed as input into the network. The block diagram of the data pre-processing process is shown in Figure [5.1]. Two procedures are conducted these are coding the non-numerical values and normalizing the data and each one is described below.

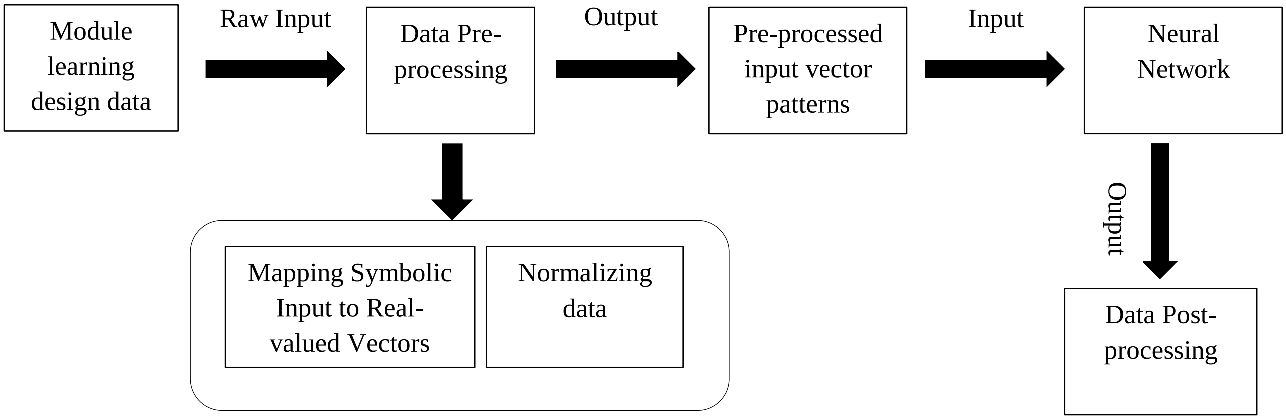


Figure [5.1]: Block diagram of data pre-processing

5.1.1 Mapping Symbolic Input to Real-valued Vectors

Information processing within ANN is numerical by nature; that is they only accept numerical input vectors and generate numerical output vectors (or a scalar). Symbolic (nominal or categorical) input variables therefore need to be transformed into a corresponding real-valued vector. There are a number of different approaches used to convert categorical into numerical feature representation; supervised lexicalized natural language processing approaches use the standard word representation which take a word and convert it to a symbolic ID. This is then transformed into a feature vector using a one-hot[12] representation features (Turian, Ratinov, and Bengio, 2010). However, the one-hot representation approach can suffer from high dimensionality if there are many values for a given categorical variable. In addition, no assumption about word similarity is given. Other approaches such the Brown Corpus (Francis and Kucera, 1979) involves that each word is associated with a tag which represents its syntactic category. Tags are assigned to bit pattern codes by frequency of occurrence in the corpus. This approach can help the training process by reflecting known similarities between symbols into their coding. The approach also uses the binary bit pattern representation which is used in previous related auto-encoder models such as PARSNIP auto-associator (Hanson and Kegl, 1987), and recursive auto-associative memory (Pollack, 1990; Voegtlin and Dominey, 2005). On closer inspection, the categorical features are found to be of ordered values, with Bloom's taxonomy behind the categories, therefore the same approach was followed by grouping the same categorical feature together representing a group of symbols of the same type or category. Each symbol is then represented as a binary vector using the Gray bit pattern coding where the hamming distance between two symbols in the category is only one bit (Black, 2004). The number of bits for each symbol is the minimum required to represent all symbols in the corresponding category. Because there are no more than 32 symbols in each group, each symbol within a group is represented by 'six bits'. This representation is used to represent each of the learning outcomes, learning objectives, teaching activities, and assessment tasks while the categorical features under the 'Subject area' were represented by 8-bits long activating one bit for each feature. However, it could be represented by three-bits to reduce the feature dimensionality. All other numerical features were represented by one single bit as real values. With each module design pattern having five different LOs, ten LObjs, twenty TLAs, and ten ATs; this adds up to a total of 288 bits used to encode all possible input symbols for the MLP neural networks as shown in Figure [5.2].

[12] One-hot encoding is a form of binary coding where each bit, or input node, represents a symbol or categorical value and only one can be active at any one time to represent a particular input value

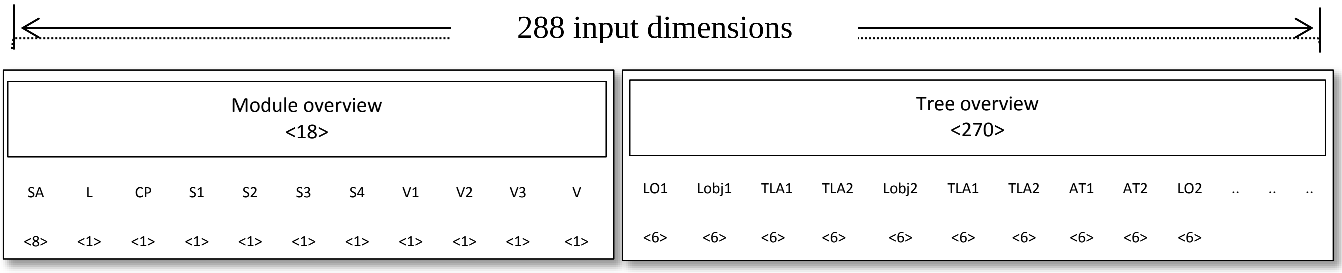


Figure [5.2]: Symbolic Input Representation (1)

Another input representation was used for the deep auto-encoder networks and DBN where the purpose was to reduce the input dimensionality and to make the input representation shorter and more effective. This was achieved by following the same approach above, however; categorical features were represented only with single bit. More clearly, each LO verb, LObj verb, TLA, and AT is given a Bloom's level number then weights are added to the Bloom's level number for each type and numerically tagged in the given context. This resulted in a total of 56 bits of dimensional vector as shown in Figure [5.3]. This can help in that each input will have some significance and thus cause a weight change resulting in learning for all weights.

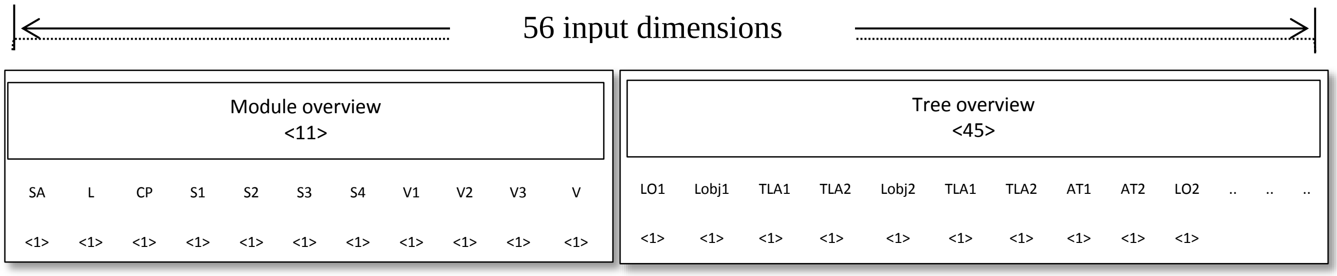


Figure [5.3]: Input representation (2)

Clearly, a post-processing approach is then needed to map the resulting vectors generated by the ANN back into the original symbolic representations for it to be read back into the metric engine and for interpretations to be made. In order to do this, each aspect of a module design has a binary code and a simple threshold function applied to the continuous activations to produce a series of binary activations that can then be symbolically interpreted. The function applies a threshold of 0.5 with any value greater than or equal to 0.5 converted to 1, and those below to 0. It then interprets the resulting binary strings by matching binary sub-strings with the different symbol representations from the symbol database by calculating the Euclidean distance to identify the appropriate learning outcome, teaching learning activity, or assessment task i.e. the symbol with the smallest Euclidean distance is the chosen symbol. The same process is applied in case of the second representation but instead of converting into binary strings, the outputs are rounded and compared to their near existing number presented in the database table.

5.1.2 Normalizing the Data

After encoding all the nominal values and that all data have been converted to numeric values, all data need to be normalised to be within a specific range in order to help to improve the performance of the network by getting good initial weights (Kevin and Keller,2005). While in theory, some studies say that it is not necessary to perform this step, however; practice has shown that data with different scales often lead to the instability of neural networks but when data are normalized, neural network training is often more efficient, which leads to a better predictor and convergence. The normalisation procedure typically consists in transforming the inputs into values in the range between 0 and 1 using the Min-Max normalization or standardizing the data between -1 and 1 using the Gaussian normalization. The function "Gaussian normalization" is used to standardize the entire data in order to yield zero mean and unity standard deviation. This is achieved using the following equation:

$$\text{X standardize} = \frac{\mathrm{x - mean}}{standard\ deviation}$$

5.2 The Training and Testing Sample

After the pre-processing procedure, the data is divided into training and testing samples. The training sample consists of 519 patterns generated from those modules with high student satisfaction scores as described in Chapter 4. This will be used to train the network by computing the error gradients and updating the network weights. The objective of the training is to achieve optimal memorization performance by producing the minimum training error. However, during the training of neural networks, over-fitting can occur which is an indication of poor generalisation (Samarasinghe, 2007). This occurs is when the training error is driven to a very small value, but when testing the network the error is large. To avoid this issue, cross-validation can be used during the training process. The idea of cross validation is to split the training set into two sets: a set of training examples to train with, and a validation set that used to measure the model's prediction. On the other hand, under-fitting is the problem when the network cannot capture the underlying structure of the data.

The testing sample consists of 102 testing patterns that are used to test the trained network. It is essential that none of the test patterns are presented in the training sample and that the set of unseen data must be exposed to the trained network in order to test its performance. As explained before the test patterns consist of input patterns from those modules with low student satisfaction scores, and when presented, the network will effectively treat these as noisy versions of patterns within the training set and therefore attempt to produce a pattern on the output layer that resembles one or more of the 'good' module designs found within the training set which is closest to the current input pattern. Because the neural networks will be trained to perform the auto-association task i.e. to reproduce a set of input patterns, the input patterns are also used as desired (or output) patterns. Figure [5.4] gives an example of this input/output pattern used for training and testing the neural network. The explanation of the design input pattern can be referenced back to Figure [4.3].

[SA_101 L_2 CP_20 S1_4.1 S2_4.3 S3_4.1 S4_4.4 V1_5.5 V2_12.3 V3_10.6 AVG_V_12.2 [LO1_102 [LOBJ1_401 [TLA1_404 TLA2_401] LOBJ2_101 [TLA1_401 TLA2_602] [AT1_601 AT2_601]]] [LO2_501 [LOBJ1_.. [TLA1_..TLA2_..] LOBJ2_.. [TLA1_..TLA2_..] [AT1_..AT2_..]]] [LO3_.. [LOBJ1_.. [TLA1_..TLA2_..] LOBJ2_.. [TLA1_..TLA2_..] [AT1_..AT2_..]]] [LO4_.. [LOBJ1_.. [TLA1_..TLA2_..] LOBJ2_.. [TLA1_..TLA2_..] [AT1_..AT2_..]]] [LO5_.. [LOBJ1_.. [TLA1_..TLA2_..] LOBJ2_.. [TLA1_..TLA2_..] [AT1_..AT2_..]]]]

Figure [5.4]: Example of Input and Output Pattern used for Training and Testing the Networks

5.3 Neural Network Experiments

A quick remainder from Chapter 3 that auto-associative networks, or as can be known as auto-encoders, are simple learning networks that aim to capture associations between input and output patterns by recalling the inputs into the outputs with the minimum reconstruction error using back-propagation or similar learning procedures. The desire is that the output needs to be as close to the input as possible, which is represented by a distance between the input and the output. The most common types of distance are the mean squared error (MSE) and the root means square error (RMSE) and will be used to measure the training performance in the conducted experiments. In following sections different neural network models utilizing multi-layer perceptron neural networks are trained to act as both shallow and deep networks for learning the process of forming associations between related patterns. In so doing the networks are trained for memorizing the features of good design patterns, where good design patterns were determined by both high degrees of student satisfaction ranging from 4 and higher and allowable range of alignment for V1, V2, and V3 identified from the good design patterns. A low MSE/RMSE is a desire for the training patterns, however for the test patterns it is not because the networks will attempt to reproduce the same poor patterns. In effect, the higher the individual pattern error, the greater the changes to those pattern and therefore the greater the information content. Therefore, the purpose here is to ascertain whether the well-trained network subsequently processes tests patterns as noisy versions of the training set and therefore seeks to modify the test pattern so that it is nudged more closely towards the training set and therefore good design patterns. In addition, to discover whether the network prefers certain types of design patterns and identifies certain changes in module design that did not appear in the training and elicits a high level student satisfaction.

Feed-forward Multi-layered Perceptron (FFMLP) with single hidden layer, deep auto-encoder networks (AEN), and deep belief networks (DBN) have been developed for this purpose. The different network models were developed due to their ability to define complex relationships between variables and discover interesting structures about the data as they tend to learn the feature representation of the data by compressing and reducing data dimensionality (Samarasinghe, 2007; Bengio, 2009). All models are constructed and trained with training set inputs and construct a predictive model that reconstructs it input in the output layer returning the least minimum reconstruction error. The training dataset is divided into a training set to train the networks and a validation set to evaluate the networks training performance and monitor over-fitting the networks. All network models will have an input layer and output layer of equal size. However, different network architectures will be considered for each NN model to obtain the best network model. The different network architectures would require different configurations of the number of hidden layers and neurons per layer, the type of the transfer function in the hidden layer and output layer, and other hyper–parameters. As different experiments will be investigated for the given problem, MSE/RMSE is used to measure the training performance of the models hence select the winning model for training set. Before discussing the various auto-associative networks, the next section briefly explains the technical work undertaken to perform the experiments.

5.4 Programming the Neural Network Models

The implementation of the neural networks models have been performed using C++ and MATALB. The C++ was the implementation of the feed-forward multi-layered perceptron (FF-MLP) trained with the backpropagation learning algorithm as defined by Rumelhart, Hinton, and Williams (1986). The implementation only supports one hidden layer and includes the Auto-association function to train the net as an Auto-encoder i.e. non-linear PCA. The code takes a batch file in the form a text file and contains the configuration filenames that need to be run. The function of the auto-association is called by the given configuration file that contains set of parameters that need to be initialised and activated. Training and testing sets are text files provided separately with input patterns. After completion of the training process, the code generates five output results appear in the results file showing the training condition. Once the training is satisfied, the mode of operation is switched to test phase and the testing configuration file is provided with the testing set file.

The implementation of the deep auto-encoder (DAE) and deep belief networks (DBN) were performed using MATLAB (R2013b) to write script files for developing the networks and performance functions for calculating the model performance error statistics such as R^2, MSE, and RMSE. The neural network toolbox in MATLAB helps create the networks by using some built-in functions that can easily help the user to specify and configure the network parameters such as the *net.trainFcn*; which, can be set to the name of the any training function used to train the network. The toolbox supports a variety of training algorithms, including several gradient descent methods (GD), scaled conjugate gradient methods (SCG), the Levenberg-Marquardt algorithm (LM), and the resilient backpropagation algorithm (Rprop). The developed networks in MATLAB are trained using the “trainFcn” and “trainParam” train functions. The trained networks are then saved by calling the functions: save (net'). When the training is complete, the network performance should be checked. Therefore, unseen data (testing) will be exposed to the network. The testing simulation process is called with the following function: *sim (net, testn);* % simulate network where testn is the testing set.

The performance function is then called to calculate and store the performance error statistics. This process is followed by using a MATLAB script where its function is to calculate the statistical results and output the degree of the coefficient of determination and the average learning the networks have achieved. The last step concludes with writing the input pattern and its predicated output pattern results along with corresponding statistical data to an Excel sheet where it can be read back into the alignment metric.

5.6 Feed-Forward Multi-Layered Perceptron (FFMLP) Trained as Shallow Auto-Encoder

Shallow auto-encoder network has simple architecture that consists of an input layer and output layer of the same size of neurons and a single hidden layer with less hidden neurons than the input/output layers. By limiting the number of hidden neurons in the hidden layer, the hidden layer will be responsible for transforming and squeezing the input into an encoding with fewer dimensions than the original one. This compression forces the auto-encoder to learn a good representation of the data and can learn some useful features of the data (i.e. features of the good design patterns) for example, if some of the input features are correlated, then the algorithm will be able to discover some of this correlations. Each module design pattern is represented as a real-valued vector of 288 dimensions where each dimension represents an input/output variable (a component of a module design) as given in Figure [5.2]. Hence, the network has an input layer and output layer of 288 neurons for all the auto-encoder networks that have been examined in this section. The created networks use the non-linear activation function and work for the case that the data lay on a non- linear surface. The module design data was fed into the designed networks and the networks were trained to recall the inputs. Thus the structure is as in Figure [5.5]

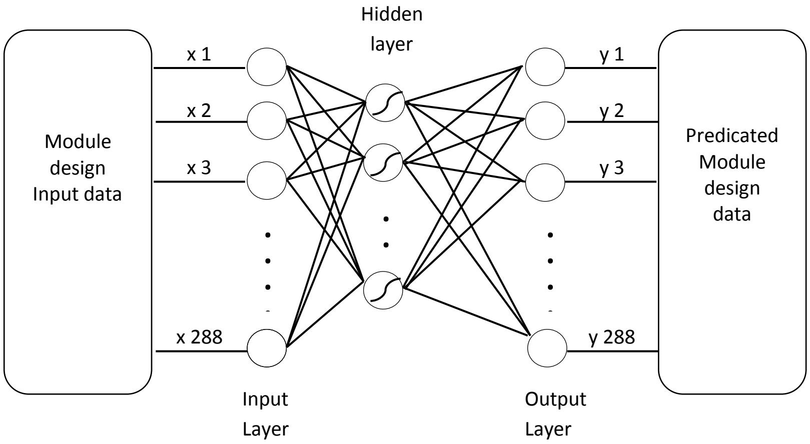


Figure [5.5]: Structure of MLP configured as shallow auto-encoder Neural Network

5.6.1 Network configurations

Different configurations of the number of hidden neurons in the hidden layer, activation functions, random initial weights, and other hyper-parameters were experimented. All literature showed that there is no standard formula for selecting the number of hidden neurons. If the number of the hidden neurons is too big, the network may suffer from over-fitting and cannot produce correct outputs when presented with unseen data. On the other hand, small numbers of hidden neurons will not help the network to converge to a solution. Different techniques and assumptions were elaborated in the literature for calculating and determining the number of neurons in the hidden layer. Researchers attempt to hit-and-trail until the best results and performance is guaranteed. Baily and Thompson (1990) suggest that the number of hidden neurons in MLP can be 75% of the number of the input neurons. Katz (1992) indicates that a typical number of hidden neurons can be found between half to three times the number of input neurons. Others proposed that the best number of hidden neurons involve hit-and trail experimentations. The experimental study begin with the number of hidden neurons in the hidden layer set to half the number of input and output neurons as proposed by Katz (1992). Then the network performance is examined and the number of hidden neurons increased/decreased based on the performance. Different numbers of hidden neurons of 140, 160, and 180 were explored with the non-linear activation function namely the hyperbolic tangent (tanh) function being used in the hidden layer and linear function in the output layer. Using a non-linear activation function in the hidden layer allows the networks to solve problems, which are out of reach of linear networks. Therefore, this can introduce a non-linearity system into the network. It was found that using the sigmoid function in the hidden layer resulted in very slow change in the error function after some iterations. In addition to that the RMSE was fluctuating and did not reach its goal. On the other hand, using the tanh function in the hidden layer, yielded great change in the network performance and good training result, therefore the hyperbolic tangent function was a good choice for the hidden layer for 0-1 encoded binary representations. In order to prevent the output to be bounded between [-1, 1], the purelin function was used in the output layer. The experiments were expanded using different learning rate types, initial learning rate, momentum term, and different range of random weights initialization to help to improve the performance of back-propagation learning and to optimize the network generalization by selecting the optimal combination. Two learning rate types were considered – fixed learning rate and the search-and-converge learning rate. Different momentum constants were used 0.9 and 0.95 as many literatures suggest that a typical value of choosing the momentum is 0.9. With each momentum constant, training sessions were carried out with learning rate fixed at 0.001 this is because the identify function being used in the output layer and this can allow the network input to blow up in some cases (causing -1.#IND) thus in order to avoid this issue the learning rate kept low. After that, for each momentum constant, training sessions were carried out again using the search-and-converge learning rate with the same initial learning rate. The purpose of the learning rate is to control the size of weight and bias changes in learning of the training algorithm. If the learning rate is very small, the network will learn very slowly; if the learning rate is too large, the model will diverge. Additionally, the momentum is used to prevent the system from converging to a local minimum or saddle point. A high momentum parameter can also help to increase the speed of convergence of the system. However, setting the momentum parameter too high can create a risk of overshooting the minimum, which can cause the system to become unstable. A momentum coefficient that is too low cannot reliably avoid local minima, and can also slow down the training of the system. Initializing the weights is important. Although the ideal initial values for weights cannot be determined theoretically, it is preferable to assign small randomly-generated positive and negative quantities as the initial weight values (Samarasinghe, 2007). The most common weight and bias initialization function is the standard normal distribution, which generates values between -1 and 1. The reason for using random initial weights is to break symmetry, while the reason for using small initial weights is to avoid immediate saturation of the activation function (Samarasinghe, 2007). In these experiments, different ranges were applied to set the initial weights within a range of +/-0 and +/-0.1.

5.6.2 Training

Training was performed using the popular and effective gradient descent method, the back-propagation learning algorithm. Training sessions were first carried out using +/-0 as initial starting weights and fixed learning type with initial learning rate value set to 0.001 to avoid the network input blowing up. Following that, the range of the starting weights changed to +/-0.1 while the learning rate type kept. Each experiment was repeated 10 times and the average RMSE was compared. It was noticed that both ranges of weights present improvement if accompanied with the appropriate number of hidden neurons. Both produced almost the same training performance averaging over 83%. Following that, training sessions with the search-and-converge learning rate type were carried out and the range of random weights +/-0 and +/-0.1 were applied as well. The training results showed that learning was improved about 4% from last session when using the search-and-converge learning rate type. Therefore, the search and converge learning rate was useful in improving the learning task.

Each training experiment was repeated 10 times (with the same architecture but different training configuration as discussed previously) to explore and account for sensitivity to the initial state determined by the randomly generated starting weights thus ensuring a network of a good accuracy. In all training experiments, the algorithm needs certain conditions upon which it can terminate, therefore all training experiments are terminated when any of these conditions occurs first: reaching the maximum number of 1000 epochs, or the RMSE training performance is minimized to the goal of 0.01.

5.6.3 Experimental results

Various network models were investigated in order to determine the optimal MLP network (i.e. the highest learning average and low RMSE). All networks were trained with the back-propagation algorithm, however; different numbers of neurons in the hidden layer were investigated. In addition, different initial starting weights and learning rate types were also investigated. Table [5.1] presents together the different network experiments and the results of these conducted experiments by displaying the average training error (AVG training RMSE) and the average generalization error (AVG test RMSE) for each model that was trained. As can be seen from the table results the success of training raises as the size of the hidden layer increases. It was found that no error improvement was gained more than 1000 epochs therefore; the training epoch was set to 1000 for all the experiments. Among all investigated MLP models, it can be seen the model 6 has produced the best training results.

The optimal number of hidden neurons appears to be 180 neurons with a momentum term of 0.95 and random weights initialization set to +/- 0.1. The search and converge learning rate used in the training process was found to aid and improve the learning task. Increasing the number of hidden neurons more than 180 gaining no further or significant improvement as can be seen from the table results.

Table [5.1]: FFMLP Training Results

Network model	**Hidden nodes**	**Transfer function in Hidden layer**	**Transfer function in Output layer**	**Initial starting weights**	**Learning rate type**	**learning rate**	**Momentum**	**Epochs**	**AVG Training RMSE**	**% Patterns learnt**
1	140	Tanh	Linear	+/-0.0	Fixed	0.001	0.9	1000	0.0379	30 %
2	160	Tanh	Linear	+/-0.0	Fixed	0.001	0.9	1000	0.0352	78 %
4	180	Tanh	Linear	+/-0.0	Fixed	0.001	0.9	1000	0.0258	83 %
5	180	Tanh	Linear	+/-0.0	Search-and-converge	0.001	0.95	1000	0.0247	86 %
6	180	Tanh	Linear	+/-0.1	Search-and-converge	0.001	0.95	1000	0.0248	87 %
7	200	Tanh	Linear	+/-0.1	Search-and-converge	0.001	0.95	1000	0.0248	87 %

5.7 FFMLPs Extended to Deep Auto-encoder Neural Networks

It is proven by many studies (Bishop, 1995; Hinton, 1992) that MLPs with one hidden layer with a sufficient number of hidden neurons can approximate and model any function. However, because of the limited capacity of that layer, the extracted features from the layer can be seen as low-level features. Thus MLP with more hidden layers can learn complex and complicated functions that can represent high-level abstractions and more effect representations as suggested by many theoretical studies most notably in (Salakhutdinov et al., 2009), (Hinton, 2006; Bengio, 2009). MLPs composed of many hidden layers are indeed an example of network models with a deep architecture as classified by (Bengio, 2007; Glorot and Bengio, 2007, 2009, 2010; Hiton 2008). This deep architecture has the potential to both improve the network generalization and to learn hierarchical representations of the input data and thus can better generalize to unseen data as well. It learns hierarchies of dependencies and features and combines them successively using several hidden layers. Based on that, deep auto-encoder structure (with three hidden layers) was investigated. The structure of all deep auto-encoders in the following experiments is as shown in Figure [5.6]. The network architecture is using bottleneck architecture and has the following layers: the mapping, bottleneck, and the de-mapping layer as hidden layers. As the task is to perform an auto-association function, both input and output layers of the networks have the same number of neurons. The mapping, bottleneck, de-mapping combinations enable the network to develop a compact representation of the training data that better model the underlying system parameters by performing non-linear principle components analysis as explained by Kramer (1992), and therefore the neurons in the hidden layers must utilize non-linear activation functions to ensure proper functioning in the network and to produce a nonlinear decision boundary via non-linear combinations of the weight and inputs (Kramer, 1992; Kerschen, 2004). In all the auto-encoder experiments performed in this subsection, the non-linear activation function of the three hidden layers utilized to be the tanh activation function and simple linear activation function (i.e. purlin activation function) being used to handle the continuous outputs in the output layer.

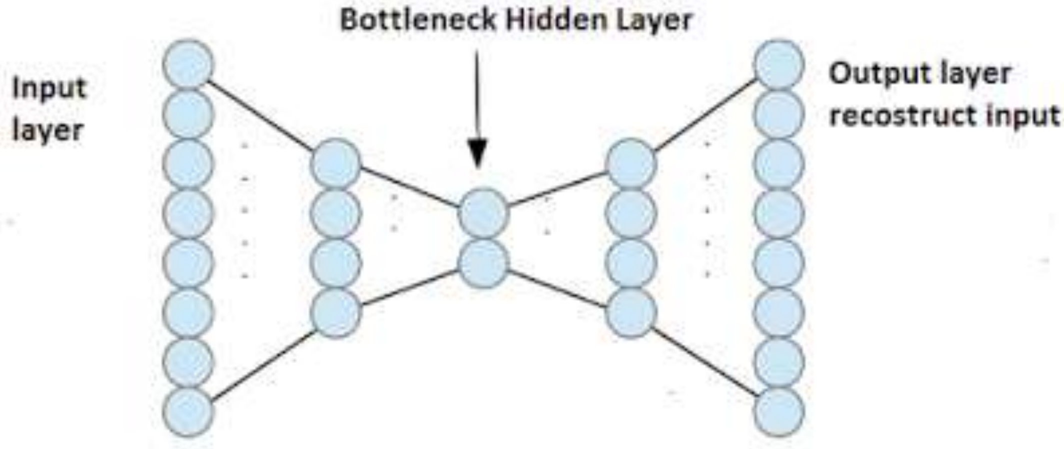


Figure [5.6]: Structure of Deep Auto-encoder Neural Network

(Source: https://metacademy.org/roadmaps/rgrosse/deep_learning/version/22)

5.7.1 Network configurations

With the number of neurons in the input layer and output layer fixed by the second method representation technique used as indicated earlier in this chapter, the network size depends on the number of neurons in its hidden layers. This makes the hidden layer neurons responsible for properties of the network such as the ability to learn and to generalise. Three hidden layers were utilized with different numbers of hidden neurons were investigated in the mapping/de-mapping layers and the bottleneck layer. The mapping and de-mapping layers can be seen as a combination of two different networks: compression and de-compression networks as described in (Kramer, 1992; Kerschen, 2004). Both of these networks meet at the bottleneck layer, which is used to perform nonlinear principle components analysis and thus forced to try to produce the output by only using small set of neurons. The method for determining the optimal number of neurons in the hidden layers is followed through the proposed example by Kramer (1991), which proposes to use greater number of neurons in the mapping and de-mapping layer than the number of neurons in the input/output layers and smaller number of neurons in the bottleneck layer than the other layers so that the ending network can have something like 3-5-2-5-3. In the conducted experiments, the input/output layers contain 56 neurons as discussed previously. The bottleneck layer constitutes of 50 neurons as starting, whereas the mapping and de-mapping layers constitutes of 58 neurons layers with sigmoid activation function (tangent sigmoid, tansig) being used in the layers. The numbers of hidden neurons were gradually increased until the network of minimum RMSE was attained. It was found that the optimal number of neurons in the hidden layers were 65 - 55 for the mapping/de-mapping and bottleneck respectively because they produced the lowest root mean square errors during training. Further increases in the hidden neurons produced higher errors. The reason of employing three hidden layer here is that no significant difference was observed in the training performance by increasing the number of hidden layers to more than three.

5.7.2 Training

The auto-encoder networks were trained using an optimised form of the popular back-propagation learning algorithm, called the scaled conjugate gradient descent (SCG) algorithm, which enables deep networks to be effectively trained (Hinton and Salakhutdinov, 2006). The algorithm updates the weight and bias values based on conjugate directions, which is scaled to avoid the line-search per learning iteration (Moller, 1993). It performs well over a wide variety of problems such as pattern recognition and pattern association as its memory requirements are relatively small, and yet it is much faster than standard gradient descent algorithms. In addition, the SCG training algorithm incorporates an adaptive learning rate and momentum parameters thus the performance of SCG is benchmarked against the performance of the other back-propagation algorithms (BP) (Rumelhart et al., 1986), which usually depend of the user dependent parameters learning rate and momentum constant as the values of these parameters are often important for the success of the algorithm.

The architecture of 56-65-55-65-56 was appropriate as we get a desired RMSE of 1.844 compared to the other architectures investigated that produced higher training errors. The training process for this architecture run using the Nguyen-Widrow layer initialization algorithm in MATLAB called INITNW, which is a layer-by-layer initialization function that initializes each layer according to the respective transfer function. The algorithm creates initial weights and bias values in order to distribute the active regions of the layers neurons evenly across the input space (Demuth and Beale, 1998; Samarasinghe, 2007).

5.7.3 Experimental results

Table [5.2] summarizes the conducted experiments where the best solution is highlighted in the table. As can be seen from the table results the success of training raises as the size of the hidden layer increases. Model 4 was further investigated by increasing the number of hidden nodes and epochs but no further learning improvement was obtained.

Table [5.2]: DAE training results

Network Model	Hidden neurons	Weights initialization	Epochs	Training RMSE	AVG Learning
1	58-50	INITNW	1000	0.2322	86%
2	61-55	INITNW	1000	0.2283	86%
3	65-50	INITNW	1000	0.2005	87%
4	65-55	INITNW	1000	0.1844	88%
5	65-55	INITNW	3000	0.1844	88%

5.8 Deep Belief Networks with deep learning

Deep Belief Networks based on Restricted Boltzmann Machines (RBMs) were investigated to form a deep multi-layer architecture with deep training. This approach is based on the observation that random initialization is a bad idea, and that pre-training each layer with an unsupervised learning algorithm can allow for better initial weights thus better training results. DBNs developed in this experiments share two additional key properties over the previous networks: the generative nature of the model, which typically requires adding an additional top layer to perform discriminative tasks, and an unsupervised pre-training step where each lower layer’s outputs are fed to its immediate higher layer as the input.

5.8.1 Network configurations

The current DBN architecture has three layers of RBMs, which is the depth of our model as graphed in Figure [5.7]. Each RBM has its own set of weights that is initialized randomly and trained independently of each other. Subsequently, the RBMs are merged together in the NN and the NN gets trained with pre-initialized weights. The RBMs layers all utilized the non-linearity tanh sigmoid function and linked such that they form a deep architecture and construct each RBM in a way they share the weight matrix and the hidden bias with its corresponding sigmoid layer. The networks are limited to a visual layer and a hidden layer, there is a connection between the layers, but there is no connection between the layers of the layer. Hidden layer units are trained to capture the correlation of higher order data in the visual layer. For setting the number of hidden neurons of the RBMs, Hinton (2010) provides a way for choosing that by estimating the typical negative $\log_2$ probability of a data-vector and then multiplying that estimation by the number of training cases, which then gives the number of neurons that is about an order of magnitude smaller of that product. This recipe is however unclear and no further details or explanation were provided, therefore it was not followed. However, most studies recommend using less neurons in the hidden layer than the input. Therefore, in these experiments, a set of RBMs were trained, each containing a different number of hidden neurons less than the input neurons. At the start of the training run, the number of hidden neurons in the first layer was half the input's neurons + 2, and so decreased slowly until the last layer. The RBMs are trained using the SGC and the parameters are set according to the algorithm's default parameters as a starting point. It was found that using the same size of hidden neurons for all hidden layers worked generally better. Increasing the number of hidden neurons larger than the input neurons does not enable the training to be performed accurately. Similarly, increasing the number of RBM layers does not provide any significantly higher accuracy. This may be data-dependent and also may be due to the parameters used which need more optimization.

5.8.2 Training

The DBNs were trained using the greedy layer-wise training approach, which is a way of initializing better the parameters of DBN by training a layer by layer, each layer is initialized as an RBM. RBM pre-training is used to obtain a faster convergence for training a deep auto-encoder and to find and produce a good initialization of the weights. The learning in DBN is performed by adjusting the interactions between variables to make the network more likely to generate the observed data. This is done through two main stages, pre-training each RBM followed by fine-tuning with back-propagation algorithm on the entire DBN to update the weights (Hinton and Salakhutdinov, 2006). The first stage involves pre-training the three layers of the RBMs independently. The first layer is trained as RBM that models the raw input data as its visible layer. Then each layer takes as input the representation learned at the previous layer and learns a new representation. After that fine-tuning is performed via supervised back-propagation algorithms on the whole DBN with the stopping criteria the same as that of deep auto-encoder networks in the previous section. As DBNs are considered nondeterministic algorithms (Hinton, 2006; Bengio, 2009), each DBN in these experiments was repeated 10 times and the average estimates have been taken to evaluate the performance of the model.

5.8.3 Experimental results

The training results show that model 3 with the given parameter values as illustrated in the Table [5.3] gives better performance in terms of the reconstruction error compared to the other architectures.

Table [5.3]: DBN training results

Network model	Model Architecture	Hidden neurons	Weights initialization	Epochs	Training RMSE	AVG Learning
1	2 RBMs	30-15	Pre-initialization of weights	500	0.6469	57%
2	2 RBMs	15-15		1000	0.6118	59%
3	2 RBMs	30-30		1000	0.3951	74%
4	3 RBMs	30-30-15		1000	0.6912	55%

5.9 Model Selection

Different experiments were conducted as seen and the root mean square error (RMSE) was used to measure the training performance of the models. The training results of the different NN models obtained is illustrated in Table [5.4], the table presents only the best results obtained from each network method. In order to measure the performance of the network models, the average learning and RMSE are used for evaluating the performance of the models. As can be seen from the table below, the model with the highest average learning accuracy of 88% is the deep auto-encoder neural network model that consists of three hidden layers with 65 neurons in the mapping and de-mapping layers and 55 neurons in the bottleneck layer. The SCG training algorithm with tansig and pureline activation function was used to train this model and the weights of each layer were randomly initialized according to the perspective activation function. It is also noticeable that model 6 is the best among all investigated MLP models as it yields the lowest RMSE with an average learning of 87%. The model consists of a single hidden layer comprising 180 hidden neurones and trained with the gradient descent back-propagation learning algorithm with a momentum term of 0.95 and random weights initialization set to +/- 0.1. The search and converge learning rate was used as it was found to aid and improve learning. On the other hand, it can be seen from the table that the AVG learning obtained from DBN models are lower than the AVG learning results obtained from the MLP and DAE neural network models. Although the DBNs were pre-trained using the greedy layer-wise fashion for getting the weights and initialization of DBN parameters, the lower RMSR of the training process achieved was 0.3951 with an average learning percent of 74%. This is considered quite a high RMSE compared to the other models which indicate that the network did not learn the problem sufficiently. The best prediction models were found to be a 288-180-288 and 56-65-55-65-56 based on back propagation algorithm with RMSE of 0.0248 and 0.1844 respectively. In order to compare the performance between the two models as different input representations being used, the RMSE for the testing set was considered. The comparison between the two models is also made by considering the statistical output measures using a paired t-test to indicate any subsequence differences due to different input representation used.

Table [5.4]: Training results of the best NN models

Network Model	Network Structure	Training RMSE	AVG Learning	Testing RMSE
MLP model 6	288-180-288	0.0248	87%	0.2947
DAE model 4	56-65-55-65-56	0.1844	88%	0.5316
DBN model 3	30-30	0.3951	74%	0.5974

As seen both networks (i.e. model 6 and model 4) indicated a successful training with an average learning above 85 % however, model 4 was slightly better compared to model 6. In terms of the generalization results, it shows that Model 6 produced a RMSE figure of 0.2974 for the test set while model 4 produced a RMSE figure of 0.5316 as shown in Table [5.4]. In effect, the higher the overall generalization error, the greater the changes to those modules in the lower 25% of student satisfaction ratings and therefore; the greater the information content. For this, further comparison of the generalization ability of both model in terms alignment and satisfaction scores was reported in Table [5.5] and presented graphically in Figure [5.8]. The table summarizes the mean values and showing the mean absolute error (MAE) for the alignment scores for the V1, V2, and V3 with the p-values. None of the p-values is smaller than the specified significance level 0.05 therefore; there is no statistically significant difference in performance among these two auto-encoder networks due the different input representation used. Thus the higher performance of the DAE trained networks was selected for testing.

Table [5.5]: Comparison of alignment scores between the MLP and DAE

	MLP	DAE	MAE	p-value
V1_mean	8.38	6.31	2.07	0.659
V2_mean	9.82	9.73	0.09	0.731
V3_mean	9.51	9.75	0.24	0.472
AVG_V	9.23	8.60	0.63	0.621

*p-value at the significant level 0.05

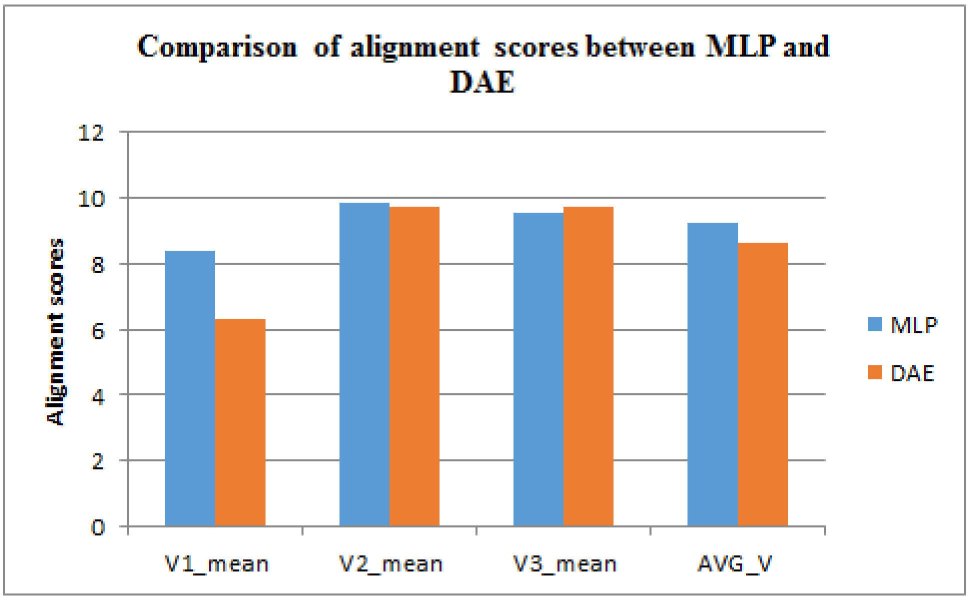


Figure [5.8]: Comparison of performance between the MLP and DAE

5.10 Summary

The underlying goal of the conduced neural network experiments is to improve the training error of the neural network model (MLP) through changes in some of the parameters, architectures, and learning of the neural network. The input layer and output layer consists of 288 neurons for all the MLP networks that have been examined while in DAE and DBN networks the input layer and output layer consists of 56 neurons due to different data input representation being used. It was shown that no significant difference was found in the predictive and generalization ability of MLP and DAE, although, the convergence speeds of DAE was higher than that of MLP. The following parameters were fixed for all the neural networks trained:

The error criterion which, used to measure the reconstruction performance of the neural network prediction to its target: “RMSE” and “MSE”.

The activation function of the hidden layers: “Tansig”.

The activation function of the output layer: “purelin”.

The training algorithm in shallow networks: “gradient descent”.

The training algorithm in deep networks: “SCG” scaled conjugate gradient descent

Other parameters that were investigated in the models as listed in Table [5.6] at the end of the chapter.

Hinton (2006, 2012); Bengio (2009, 2012) and others have reported that deep learning has advantages due to their learning methodology by combining unsupervised pre-training and supervised fine-tuning, which usually gives better generalization than pure supervised learning from a purely random initialization. In this study, no significant improvement in the network learning was found when using deep learning. It was found also based on the conducted experiments that training with the scaled conjugate-gradient method with random weight initialization is much faster than the standard backpropagation algorithm. Using the scaled conjugate-gradient method also avoids the need to search for proper settings of parameters such as the learning rate and momentum. However, the appropriate number of hidden neurons was needed to be determined. It should be noted that DBNs have proven their usefulness and ability as a modelling sequences as in Busseti et al. (2012); Boulanger-Lewandowski (2014); however, It is inefficient and unproductive within our problem's region. The reason for this inability could possibility be that DBNs need more optimization and further generalization improvement, which can be seen as future works. By summarising the different network experiments and selecting the best model, the next chapter will analyse and evaluate the output results obtained from the selected winning model for training set and extract the underlying design principles identified for transforming a poor designed module into a good module design.

Table [5.6]: Network parameters used and their values

Parameters	**Values**
MLP	
Number of neurons in the hidden layer	140, 160, 180 , 200
Learning rate type	Fixed, Search-and-converge
Momentum	0.9, 0.95
Activation function of the hidden layer	Sigmoid, Tanh
Ranges of weights initialization	+/-0.0, +/-0.1
DAE	
Number of neurons in the hidden layer	58-50, 61-55, 65-50, 65-55, 65-55
DBN	
Number of neurons in the hidden layers with 2RBMs layers	30-15, 15-15, 30-30
Number of neurons in the hidden layers with 3RBMs layers	30-30-15

CHAPTER 6: Results and Discussion

6.1 Analysing the Results of the Selected NN Model

This section presents the testing results and the numerical analysis conducted to analyse the output generalization results produced form the best neural network obtained (i.e. model 4 denoted as 56-65-55-65-56). The network produced a training error of 0.1844 and higher learning accuracy of 88% which, indicates that its performance outperform over the other networks as presented in chapter five. Hence, the network was selected and used for testing the unseen dataset associated with low student satisfaction scores to perform a key auto-association task. The network was shown 102 new design patterns which were distinct from the 519 design patterns the network learned. The generalization (test) performance for this network is shown in Table [6.1]. The network recognised a total of 5 patterns within the 0.03 error threshold i.e. indicating input patterns were identical to the output patterns and therefore matched a module within the training set. The remaining 97 test patterns generated higher pattern errors suggesting that substantial changes to the input pattern had been generated on the output layer. These differences can be interpreted as changes to a test pattern (a module with low student satisfaction scores) that have been informed by a knowledge base of good module designs (as found in the training set) and therefore if such changes were made would likely result in increased student satisfaction and hopefully alignment.

Table [6.1]: Auto-encoder network (model 4) test performance

RMSE	Number of patterns within error limit	Number of patterns outside error limit	% Pattern Generalisation
0.5316	5	97	3.921%

Since the substantial interest focuses on the effect of students satisfactions on predicting good alignment scores, it is natural to study the effect of the overall student satisfaction scores with respect to each of the alignment scores V1, V2, V3 and overall module alignment and vice versa. Various statistical analyses were carried as it has been done in Chapter 4. Descriptive analyses were conducted to present the state of the testing set before and after changes applied by the neural network and to illustrate the average score of predictor variables V1, V2, V3 and student satisfaction. Correlational analysis was performed to identify the relationship found between the three variables of V1, V2, and V3 and student satisfaction. The coefficient of determination and R-squares were also performed to investigate whether the three predictor variables significantly predict student satisfaction. And finally self-organizing map was used to visualize an overview of the changes made to test patterns before and after.

6.1.1 Descriptive Analysis

Two tables are illustrated below. Table [6.2] compares and summaries the average scores of V1, V2, and V3 and student satisfaction scores for the test set before and after the changes that have been applied by the network and Table [6.3] compares the average scores for test set before and after with respect to the training dataset. It appears from the table that the averages of the V1 and V3 alignment scores have been decreased whereas the average of the V2 alignment scores has been increased from 7.59 to 9.47, which means that there is an increase in the alignment scores of 24.8 %. This increase in the mean of V2 alignment scores can be interpreted as strong changes the network made for raising the alignment values by suggesting higher level of Bloom’s for the given TLAs. On the other hand, the decreases in the V1 and V3 alignment scores indicate some changes that result in moving down the level of Bloom’s for the given learning outcomes (LOs) and the assessment task (ATs). In general these changes have led to change in student satisfaction average as shown in the table rising from 3.3 to 3.8. Analyses of variance indicate that there are indeed significant differences in the student satisfaction means since the resulted p-value given was less than the 0.05 significance level. This implies that student satisfaction scores were noteworthy affected with such changes generated in the design patterns in a way that has raised the satisfaction level.

Table [6.2]: Average scores of V1, V2, V3 and S for Test set BEFORE and AFTER

Test Set BEFORE		
Variable	**Mean**	**SD**
V1	9.60	5.24
V2	7.81	2.39
V3	10.15	3.14
AVG_V	9.19	2.75
Student Satisfaction	3.3	0.31
Test Set AFTER		
V1	6.31	2.82
V2	9.73	1.99
V3	9.75	2.98
AVG_V	8.60	1.51
Student Satisfaction	3.8	0.71

Table [6.3]: Differences between averages of V values of the training set and test set BEFORE applying auto-encoder network and AFTER applying the network

Training set				
Variable	**Mean**	**SD**	**Min - Max**	**Range**
V1	7.91	3.21	4.8 – 19.8	15.0
V2	9.26	2.39	4.3 – 16.3	12.0
V3	10.45	2.74	4.3 – 17.6	13.3
AVG_V	9.21	1.62	4.7 – 13.5	8.8
Student Satisfaction	4.52	0.28	4.0 – 5.0	1.0
Test set Before				
Variable	**Mean**	**SD**	**Min - Max**	**Range**
V1	9.60	5.24	3.0 – 21.2	18.2
V2	7.81	2.39	3.7 – 16.4	12.7
V3	10.15	3.14	4.0 – 16.7	12.7
AVG_V	9.19	2.75	4.4 – 14.3	9.9
Student Satisfaction	3.36	0.31	1.8 – 3.7	1.9
Test set After				
V1	6.31	2.82	1.8 – 14.8	13.0
V2	9.73	1.99	5.36 – 16.1	10.7
V3	9.75	2.98	3.8 – 16.6	12.8
AVG_V	8.60	1.51	5.73 – 13.3	7.6
Student Satisfaction	3.82	0.17	3.5 – 4.1	0.6

The acceptable range values obtained from the training data set, as presented in chapter 4 in Table [4.15], were applied separately to V1, V2, V3 and V of TestSet (before) and TestSet (after). Table [6.4] below shows the number of items data that fall within the acceptable range and their respective percentages for both TestSet (before) and TestSet (after). In general, there are more data within the acceptable range after the network than before with a total of 197 (before) and 266 (after) – an increase of 69 data points moving closer to the mean (central) value after the network. This increase occurs mainly in V1 and V2 with an extra of 23 data moving into plus or minus one SD in the case of V2. This is noted by 78% data within $\pm 1\ SD$ for TestSet (after) as against 55 % for TestSet (before).

Table [6.4]: Number of data within $\pm\ \mathbf{1}\ \boldsymbol{SD}$

	Before		**After**	
	Number (X)	**% (X)**	**Number (X)**	**% (X)**
V1	37	36%	58	56%
V2	57	55%	80	78%
V3	63	61%	61	59%
AVG_V	40	39%	68	66%

In order to identify whether or not there are significant differences between the average values of V1, V2, and V3 alignment before and after applying the auto-encoder, a two-tailed t-test was applied for the average alignment scores using the training set and test set (before) and, thereafter, using the training set and test set (after), and this is given in Table [6.5]. It was noticed previously from Table [4.13] in Chapter 4 that there was a statistically significance difference between the mean of V1 (before) compared to the mean of V1 of the training set since the p-value was small at the significance level of 0.05. Although that the changes applied by the network to V1 alignment scores have moved more than 20 data points from V1 (before) into plus or minus one SD i.e. within the acceptable range of V1, the p-value in the t-test table below suggests that there is still a statistically significant difference between V1 (after) and V1 of the training set. This explains that the network generalization on associating learning outcomes and learning objectives was different from what was shown. The t-test table below also shows that there was a statically significant difference between the mean of V2 (before) and the mean of V2 of the training set where the p-value < 0.05, however; after applying the network changes to 23 data point in V2, the t-test results indicate no significant difference between the mean of V2 of the test set (after) compared to the mean of V2 of the training set. This shows that the network had succeeded in bringing the mean of the alignment values close to the mean of the training set than it was before. In other words, the network attempted to edit either the Bloom level of the objective or the type of the associated TLA in the design patterns to bring the V2 alignment values close to the allowable ranges. In case of the V3 alignment scores the t-test results applied between the mean of V3 (before) and the mean of the V3 of the training set showed that there is no statistically significant difference between the means since the calculated p-value was greater than the significance level of 0.05. The case was the same after applying the auto-encoder changes to the V3 alignment scores, which illustrates that there is also no statistically significant difference between the means as shown in the t-test table. This can be explained as hardly such changes were taking place in V3 which may be due to no significant difference were initially found between the training patterns and the test patterns. However, the drop in the mean of V3 alignment scores is likely to be because of the changes applied to the learning outcomes in V1. Furthermore, the differences between the test set (before) and the test set (after) were further analysed using the pried t-test which shows that the t-stat is significantly higher than the t-critical in all cases except V3. As a result, it can be noted that there is a significant difference in the two paired groups in each of the V1 and V2 but not V3. The t-test for this can be found in **Appendix [G]**.

Table [6.5]: T-test for difference between mean V values of training set and test set BEFORE and AFTER applying auto-encoder

V1, V2, and V3 Alignment – Training set versus Test set (Before)						
	V1(Trg)	V1(B)	V2(Trg)	V2(B)	V3(Trg)	V3(B)
Mean	7.91	7.71	9.26	8.79	10.45	10.28
Variance	10.32	3.37	5.71	1.59	7.51	2.90
t Stat	0.583		2.333		0.703	
P-value	0.561		0.021		0.483	
V1, V2, and V3 Alignment – Training set versus Test set (After)						
	V1(Trg)	V1(A)	V2(Trg)	V2(A)	V3(Trg)	V3(A)
Mean	7.91	7.13	9.26	9.50	10.45	10.21
Variance	10.32	3.49	5.71	1.37	7.51	2.61
t-Stat	2.739		1.416		0.991	
P-value	0.007		0.158		0.323	

* Significant at the $p < 0.05$ level.

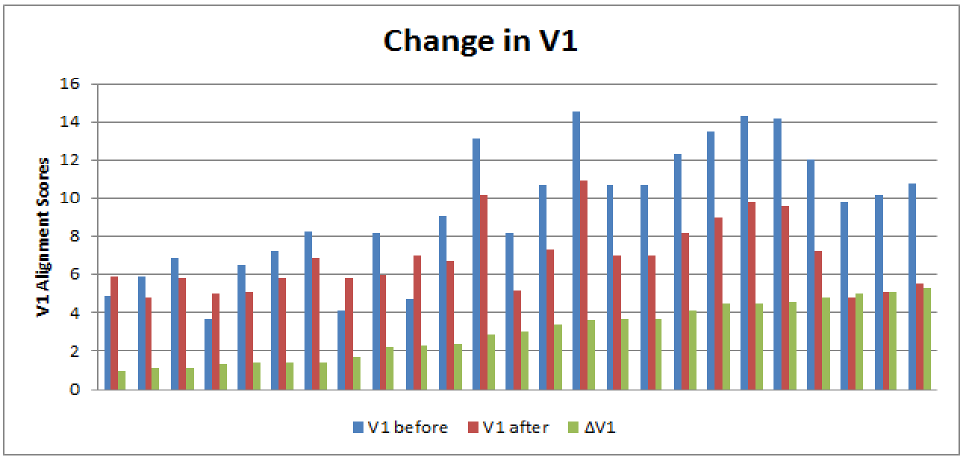
The effect size d was taken into account to further understand the changes, which happened in V1, V2, and V3. The effect size is a simple way of quantifying the difference between two groups or the changes before and after measure (Cohen, 1988). According to Cohen, this can be calculated using the equation below where M_1 and M_2 are the means of the two groups (before and after) and SD the average of their standard deviations.

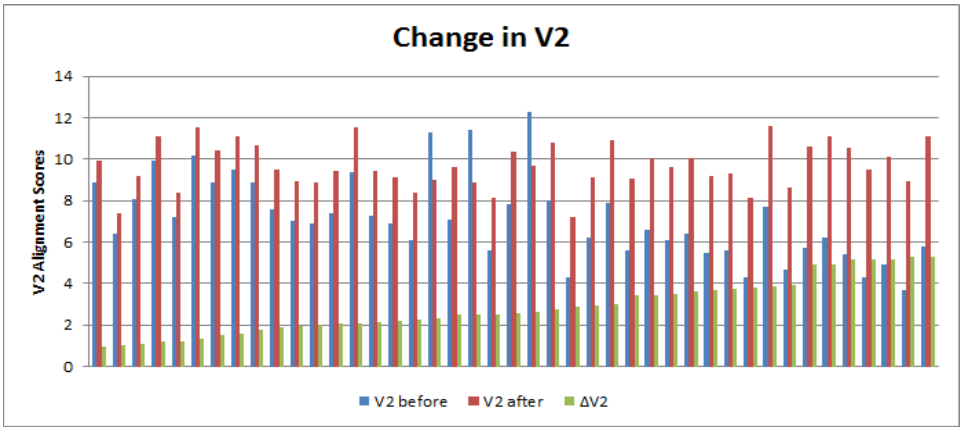
$$d = \frac{|M_1 - M_2|}{SDpooled}$$

Table [6.6] shows the effect size summary where an absolute difference of 1 or more is taken as a big difference when a trend in the scores is evaluated. According to Cohen's d in interpreting the result of the effect size, it shows that a d = 1.0 means that that the two groups' means differ by one standard deviation and the effect size is considered large; a *d* = 0.5 indicates that the two groups means differ by half a standard deviation which represents a medium effect size; and an effect size of 0.2 or less can be considered a small effect. In reference to the result table below it shows that there is about one standard deviation difference between the two sets for V1 and one and a half standard deviation difference in V2 representing a 'large' effect size while the effect size is medium for V3 as d = 0.5. Figure [6.1] shows plots of Vs comparing the alignment scores before and after for the test set where the difference is equal to or more than 1.

Table [6.6]: Effect Size based on Cohen (1988) Index using Pooled Standard Deviation

Case	**Num (X)**	**% (X)**	**Mean**		**SD**		**ABS Diff**	**D**	**Level**
			Before	**After**	**Before**	**After**	**$\lvert M_B - M_A \rvert$**		
V1	25	24%	9.38	6.86	3.32	1.80	2.52	0.98	Large
V2	46	45%	7.13	9.67	2.02	1.09	2.54	1.53	Large
V3	30	29%	10.88	9.84	2.55	1.42	1.04	0.52	Medium





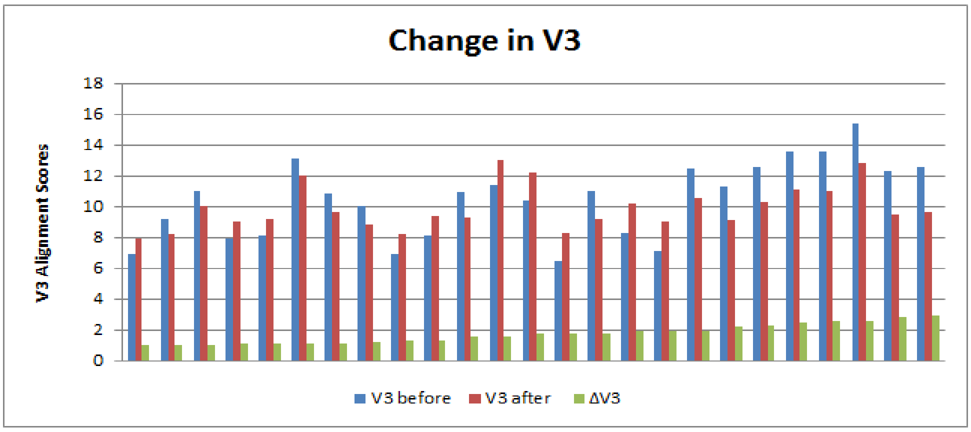


Figure [6.1]: Graphs of Vs alignment scores before and after where the difference is >=1

The following table, Table [6.7], summarizes the average results and shows the direction of changes in the alignment scores for those test patterns that came within the $\pm 1\,SD$ after applying the changes made by the auto-encoder network.

Table [6.7]: Direction of changes in the alignment scores for test patterns

Test set before compared with Test set after			
	Before		**After**
V1	9.38	↓	6.86
V2	7.13	↑	9.67
V3	10.88	↓	9.84
S	3.3	↑	3.8

To better understand the changes applied to V1, V2, and V3, this required to understand what happened to the Bloom's level for the LOs, LObjs, TLAs, and ATs. First, a comparison table was conducted to understand the trend in the Bloom's levels of the outcomes and to compare the distribution of the Bloom levels of the learning outcomes in the test set (before) and test set (after). As indicated below in Table [6.8], Bloom level 3 (Application) is the most frequent level in the test set (after), with 49% of the learning outcomes. This is followed by Analysis with 24% and Understanding with ≈ 20%. The table also shows the distribution of the Bloom's levels of the learning outcomes before and after. The table suggests a decrease in the knowledge level, an increase in the understanding, application, and analysis level, but a decrease again in the Synthesis and evaluation levels of the learning outcomes.

Table [6.8]: Comparison of Bloom's Levels of Learning Outcomes in Test Set (before) and (after)

Bloom's level	**Before**		**After**		
	N	*%*	*N*	*%*	
Level 1: Knowledge	25	5.4	1	0.2	
Level 2: Understanding	63	13.7	91	19.8	
Level 3: Application	189	41.1	226	49.2	
Level 4: Analysis	76	16.5	112	24.4	
Level 5: Synthesis	70	15.2	29	6.5	
Level 6: Evaluation	36	7.0	0	0.0	
Total	459		459		

Second the frequency analysis was run for the test set (after) to identify the most common parent-child relationships that have been suggested by the network. This is shown in the following tables below where a brief note is given under each table for a quick comparison of how was the relationship before. The new formed relationships were highlighted and italicized and the full discussion is given at the end of the analysis.

Table [6.9]: V1 (after) Frequency Relationships Table

Lo/Lobjs	**Lo(1)**	**Lo(2)**	**Lo(3)**	**Lo(4)**	**Lo(5)**	**Lo(6)**
Lobj(1)	**0(0%)**	**10(6.0%)**	**0(0%)**	**1(0.44%)**	**0(0%)**	**0(0%)**
Lobj(2)	**1(50%)**	**19(11.4%)**	**70(15.0%)**	**28(12.5%)**	**4(3.0%)**	**0(0%)**
Lobj(3)	**1(50%)**	**95(57.2%)**	**183(39.3%)**	**67(30%)**	**45(34.0%)**	**0(0%)**
Lobj(4)	**0(0%)**	**34(20.4%)**	**136(29.2%)**	**84(37.5%)**	**53(40.0%)**	**0(0%)**
Lobj(5)	**0(0%)**	**8(4.8%)**	**70(15.1%)**	**38(16.7%)**	**29(21.9%)**	**0(0%)**
Lobj(6)	**0(0%)**	**0(0%)**	**6(1.2%)**	**6(2.6%)**	**1(0.7%)**	**0(0%)**

For comparison, Table [4.9] shows that the V1 (before) relationship table as: LO(1) associated with Lobjs(2),(3), ***LO(2) associated with Lobjs(2),(4)***, ***LO(3) associated with Lobjs(3),(5),*** LO(4) associated with Lobjs(3),(4), LO(5) associated with Lobjs(3),(4), ***LO(6) associated with Lobjs(3),(4).*** Thus, V1 (after) has higher objectives associated with level 2, lower objectives associated with level 3, 4, and 5.

Table [6.10]: V2 (after) Frequency Relationships Table

Lobj/TLAs	Lobj(1)	Lobj(2)	Lobj(3)	Lobj(4)	Lobj(5)	Lobj(6)
TLA(1)	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**
TLA(2)	**6(21.4%)**	**65(16.1%)**	**39(5.3%)**	**18(3.5%)**	**0(0%)**	**0(0%)**
TLA(3)	**7(25.0%)**	**98(24.3%)**	**139(18.9%)**	**79(15.4%)**	**33(10.1%)**	**2(4.5%)**
TLA(4)	**12(42.8%)**	**53(13.1%)**	**157(21.0%)**	**125(24.4%)**	**105(32.4%)**	**26(59.0%)**
TLA(5)	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**
TLA(6)	**3(10.7%)**	**187(46.4%)**	**401(54.7%)**	**290(56.6%)**	**186(57.4%)**	**16(36.3%)**

For comparison, Table [4.10] shows that the V2 (before) relationship table as: ***Lobj(1) associated with TLAs(2),(3), Lobj(2) associated with TLAs(2),(4), Lobj(3) associated with TLAs(2),(4), Lobj(4) associated with TLAs(2),(4), Lobj(5) associated with TLAs(3),(4),*** Lobj(6) associated with TLAs(4),(6). Thus, V2 (after) has higher TLAs associated with level 1, higher TLAs associated with level 2, higher TLAs associated with level 3, 4, and 5.

Table [6.11]: V3 (after) Frequency Relationships Table

Lo/ATs	Lo(1)	Lo(2)	Lo(3)	Lo(4)	Lo(5)	Lo(6)
AT(1)	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**
AT(2)	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**	**0(0%)**
AT(3)	**0(0%)**	**9(5.9%)**	**29(6.3%)**	**9(4.0%)**	**2(1.5%)**	**0(0%)**
AT(4)	**0(0%)**	**44(28.9%)**	**184(40.0%)**	**84(37.5%)**	**19(14.6%)**	**0(0%)**
AT(5)	**0(0%)**	**60(39.4%)**	**133(28.9%)**	**64(28.5%)**	**64(49.2%)**	**0(0%)**
AT(6)	**1(100%)**	**39(25.6%)**	**113(24.6%)**	**67(29.9%)**	**45(34.6%)**	**0(0%)**

For comparison, Table [4.11] shows that the V3 (before) relationship table as***: LO(1) associated with ATs(4),(6),*** LO(2) associated with ATs(4),(6), LO(3) associated with ATs(4),(5), LO(4) associated with ATs(4),(6), ***LO(5) associated with ATs(4),(6),*** LO(6) associated with ATs(4),(5). Thus, V3 (after) has higher assessment type to assess level 1, higher practical assessment types to assess 5.

It is worth mentioning here that the network has no access to semantic information and relies on deterministic processing and symmetric forward and backward association to learn the identity function so that it learns to become a perfect memory of good design patterns. With its task to perform the pattern association, the most common relations were generated for each V1, V2, and V3. The frequency results from Table [6.9] above suggest that there is a prevalent pattern in how the network has associated the learning outcomes in their units with learning objectives appearing to be focused on the midline of the table where it attempts to use mainly the application and analysis levels in relation to the comprehension, application, analysis, and synthesis learning outcomes. For example, the network was given the relation: ***LO(2) Lobj1(2) Lobj2(4)***, the network edits the relation and corrects the Bloom level of the first objective to elicit a higher Bloom level and type and produces: ***LO(2) Lobj1(3) Lobj2(4)*** to form a correct and compound relation. It can be seen also that the network did not make any preferences or associations with 'evaluation' learning outcomes. In fact the network transformed all learning outcomes with Bloom level 6 into Bloom level 5. This transformation has been through lowering the levels of the learning outcomes one level down.

The network demonstrates a common preference for associating higher-level activities with the differing learning objectives as seen in Table [6.10]. Initially the network was presented with design patterns where the structure of the TLAs appears different from what the network actually learnt, this makes the network to perform such a syntactic disambiguation, this when a set of design patterns are given having more than one possible structure. In this case, the network corrects the design patterns and inserts the correct types of TLAs to produce well-formed design patterns.

Although the frequency table of V3 relationship illustrated in Table [6.11] looks much similar to the frequency table of V3 test set (before) in Chapter 4, a couple of LO/ATs relationships were detected after applying the network changes. The major change was in linking higher assessment tasks (Bloom's 6) with the low level learning outcomes (i.e. knowledge level-Bloom's 1). Also linking higher types of assessment tasks (Bloom's 5) for assessing the outcomes in the synthesis levels (Bloom' 5), for example, higher practical assessment types. This behaviour of the network stems from its recognition to what it has seen before and this is an expected behaviour from auto-associative networks in which they act much more as a pattern store (Hanson and Kegl, 1987). An analysis of the unchanged patterns in the LO/AT relationships showed that the network was typically able to match the closet assessment task resulting in the transformation from Exams (Bloom's 6) as assessment tool to Essay Examinations (Bloom's 6), which will be discussed later in this section.

6.1.2 Correlational Analysis

H0: There is no correlation between V alignment score and S student satisfaction score.

In order to investigate and identify the relationships that were performed between good module designs according to student satisfaction, the data were subjected to correlation analysis. The Pearson correlation coefficients r were run to look for relationships between student satisfaction and each of that the following variables V1, V2, V3, and AVG_V respectively. This is summarized in Table [6.12] which presents an overview of the correlations that were found to exist between each of the alignment scores and satisfaction score. The table shows that again student satisfaction scores with V2 alignment scores (r = 0.645, $p < 0.05$) is relatively highly correlated in comparison with V1 and V3 alignment scores (r = 0.504, $p < 0.05$) (r =0.415, $p < 0.05$). These findings are more analysed below where each V is hypothesised against the above null hypothesised and supported by a correlation scatterplot.

Table [6.12]: Correlation Analysis between the student satisfaction and V1, V2, and V3 alignment scores

	Pearson Correlation	*P-value*	*R-square*
V1 and S	0.504	< 0.05	0.254
V2 and S	0.645	< 0.05	0.417
V3 and S	0.415	< 0.05	0.172
AVG_V and S	0.662	< 0.05	0.438

*Correlation is significant at the 0.05 level

<u>Student satisfaction with V1</u>

Table [6.12] shows that a comparison was made using Pearson's r on the relationship between V1 alignment scores and student satisfaction. The result of the comparison shows a moderate correlation (r = 0.504, significant at the 0.05 level) thus we can determine that there is a correlation between the two variables. However, to determine the significance of this relationship, the coefficient of determination, or R-square, was then calculated on this correlation by squaring the Pearson's r coefficient. This gives us a measure of how important the correlation is, because even if there is a correlation, if it only explains a small amount of the variability then it might not be a very important or strong correlation (Hinton, 1995). The resulting R-square is 0.254, indicating that 25% of the total variation in satisfaction can be explained by variation in V1. The scatterplot is shown in Figure [6.2]

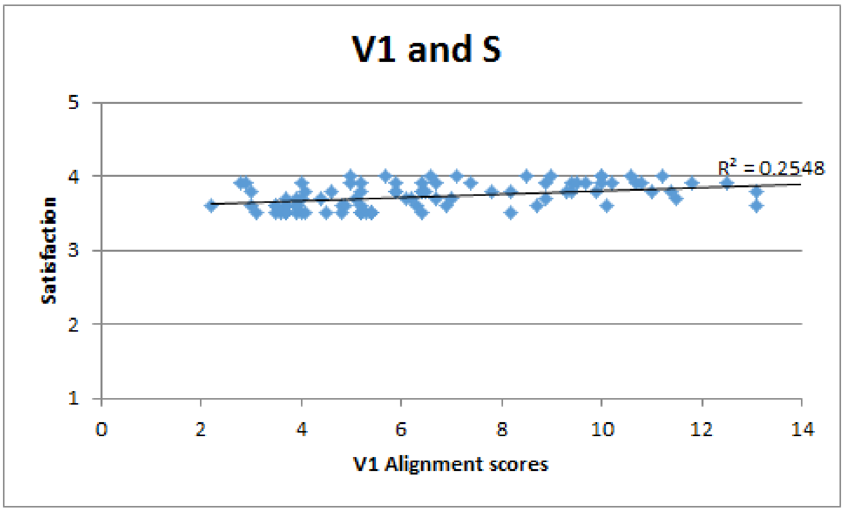


Figure [6.2]: Scatterplot for Relationship between Satisfaction and V1

Student satisfaction with V2

In case of V2, The result of the comparison shows a significant high correlation (r = 0.645, significant at the 0.05 level) and the resulting coefficient of determination calculated is 0.417, indicating that 42% of the total variation in satisfaction is derived from the V2 that is the relation between learning objectives and the type of TLAs used. It seems that when the V2 alignment score is increased, that is more high level cognitive activities are suggested, the level of student satisfaction is likely to increase as well. The scatterplot of this relation is shown in Figure [6.3]

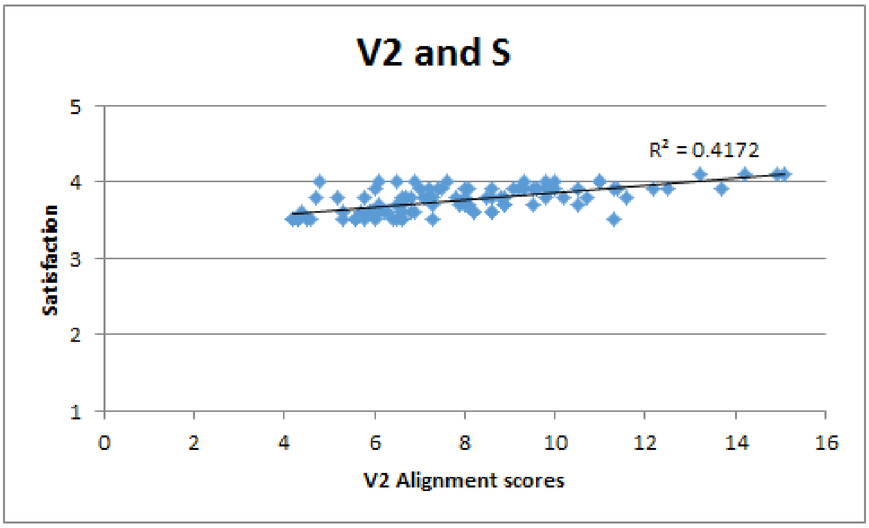


Figure [6.3]: Scatterplot for Relationship between Satisfaction and V2

Student satisfaction with V3

The result of V3 with respect to satisfaction score shows that there is a correlation formed between the two variables with (r = 0.415, significant at the 0.05 level) and the resulting coefficient of determination calculated is 0.172, indicating that 17% of the total variation in satisfaction is derived from V3. The scatterplot of this relation is shown in Figure [6.4].

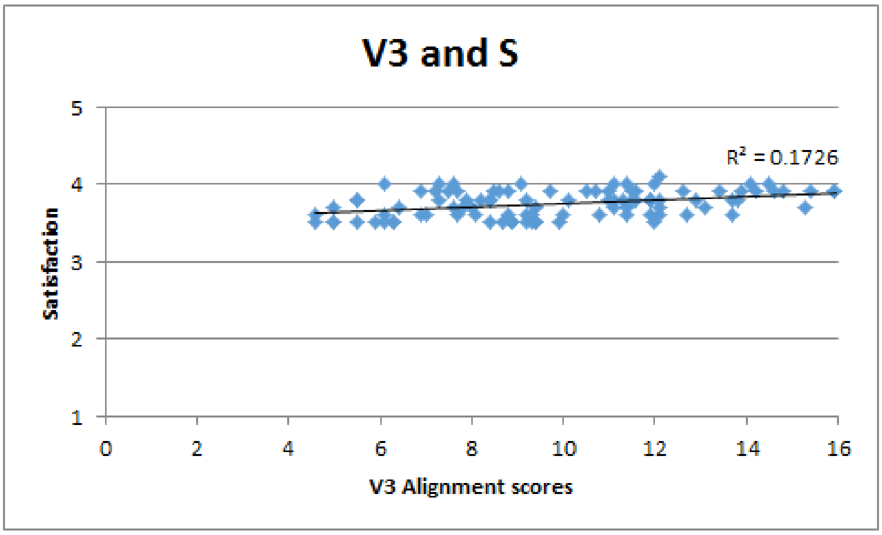


Figure [6.4]: Scatterplot for Relationship between Satisfaction and V3

Student satisfaction with overall alignment

As can be seen that there were some correlations with different degrees between the different alignment scores and satisfaction as a result the relation between the overall alignment and satisfaction found to be positively high (r = 0.662, significant at the 0.05 level) and the resulting coefficient of determination calculated is 0.438, indicating that 44% of the total variation in satisfaction is derived from the overall module alignment. This is well supported by the scatterplot which is shown in Figure [6.5].

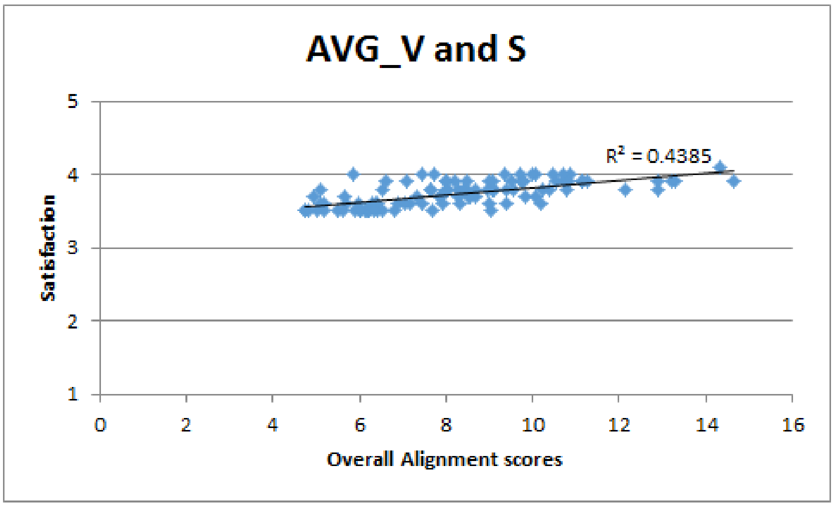


Figure [6.5]: Scatterplot for Relationship between Satisfaction and Overall Alignment

To conclude, the correlation analysis that was performed in this section to investigate the relation between the three predictor variables V1, V2, and V3 and student satisfaction has revealed the degree of association between the module designs and student satisfaction. This allows us to reject the null hypothesis and confirm the existence of a relationship between the variables. This was supported with multiple scatterplots to demonstrate the relationship between satisfaction and module design features (V1, V2, and V3). The result also revealed that V2, which measures the degree to which a given learning objective is aligned with the dominant TLAs in the module, is the most significant predictor of student satisfaction. Approximately 42% of the variance in student satisfactions was derived from the type of TLAs being used. Student satisfaction with V1 and the type of learning outcomes and learning objectives was the second significant predictor of satisfaction (25%). It was found that assessment tasks got the lowest percentage in predicting student satisfaction. It is well known that assessment tasks directly link to what students have learned and provide a way for involving more about what they are doing and not just what they know, however; it can be interpreted that the kind or type of assessment does not necessarily affect the level of satisfaction compared to the types of TLAs used during the educational process or the learning outcome verbs introduced to help students better understand what is expected of them during the educational process. The impact of TLAs on students constitutes the largest proportion as TLAs are considered one of the most important elements of education. There are substantial studies related to the link between the student satisfaction and the effective teaching methods. The choice of appropriate and effective teaching methods helps to improve the learning of the students by creating interest in the subject and the enthusiasm to learn and developing the creativity sense in the students within themselves. Highly motivated students also tend to be more satisfied with their education (Jones, 2008; Roebkin, 2007) and this is achieved by using more teaching methods that engage the students with their learning. Previous studies show that students' academic success and satisfaction relies on certain features of learning environments, particularly on high level activities such as group work activities and problem-solving exercises (Gokhale, 1995; Chalmers, 2008;). This is quite consistent with the correlation result of V2 that suggests that whenever the alignment score is increased by using high-level activities, the student satisfaction score will increase as well.

6.2 Visualizing Data with Self-Organizing Map (SOM)

The underlying representation or 'hypothesise' formed by the network has shown that there are significant correlations obtained between the alignment scores and student satisfaction in the test pattern after applying the network changes. In order to further analyse this correlation and identify what design changes were made for transforming V1, V2, V3 and student satisfaction, SOM has been used as a visualization method for identifying patterns and clusters in the data. It expresses the data in a way that similarities and differences are more perceptible. SOM is an unsupervised learning algorithm developed by Kohonen (1995). The algorithm breaks down high-dimensional data into simplified abstractions producing a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map (Kohonen, 1998; Samarasinghe, 2006). The resulting map avails itself readily to visualization, and thus the distance relations between different data items can be illustrated in a familiar and natural manner which makes SOM a powerful visualization tool. The main principle of the SOM is the application of competitive learning in which for every input vector, nodes compete with each other to see which one of them is the most similar to that particular input vector. The Euclidean distance function is used to measure the level of similarity between an input pattern and each weight vector (and thus cluster unit). The node on the SOM whose weight vector is closest to the current input vector is deemed to be the 'winning node' for that input pattern. The associated weight vector is then subject to a simple learning rule to move it towards the winning input pattern. Thus similar data items are located close to each other and dissimilar data items are farther a part in the map display. This natural groupings help to identify clusters within the datasets and to reveal what features the members of a cluster have in common and to visualize the data in such a way that both similarities and differences can be distinguished. The following subsections will discuss the map initialization and training followed by visualizing the comparison of SOM patterns for set A (test pattern after changes made by the auto-encoder network) and set B (test patterns before changes made by the auto-encoder network).

Map Initialization and Training

The SOM implementation was made with the SOM_Toolbox which is function package for MATLAB implementing the Self-Organizing Map (SOM) algorithm (Vesanto, 1999). The toolbox provides some functions and default setting/parameters for creating, initializing and training SOMs using a range of different kinds of topologies. The toolbox is a free software and relatively easy to use. The mathematical details of the SOM algorithm can be found in Kohonen (1995) Cottrell (1998) and will not be considered here. However, the description for applying SOMs will be given below.

By default linear initialization, batch training algorithms, and Gaussian neighbourhood functions were applied along with the following function SOM_MAKE that creates and trains the SOM with default parameters. The members of the training dataset were presented iteratively and randomly to a SOM of 9x9 map size. The map size was generated subject to the adaptation process mentioned in the literature (Kohonen, 1996; Bação, 2008). Several runs with different map sizes were investigated to ensure that dissimilar patterns were not forced to cluster together. The training is done in two phases respectively: topological ordering of the weight vectors by training with large (initial) neighborhood radius and large (initial) learning rate, then weights are fine-tuned with small radius and learning rate (Kohonen, 1996; Bação, 2008). After a total of 10000 iterations, the map shown in Figure [6.6] was obtained using the SOM_SHOW function. The map basically is attempting to visualize the topology of the SOM. The figure shows the neuron locations in the topology, and indicates how many of the training data are associated with each of the neurons (cluster centres). The maximum number of hits associated with any neuron is 18. Thus, there are 18 input vectors in that cluster. The distribution was clustered in many areas, which indicates a degree of separation between the patterns. A sammon map is also shown to graphically illustrate the clusters and distances between them as shown in Figure [6.7]. The map represents each cluster node with the most representative pattern i.e. the closest pattern, which has the smallest Euclidean distance to the cluster node is displayed. It is apparent how the set of nodes are displayed with position and colour indicating clusters of data. Design patterns having similar design features are arranged close to each other and the distance between represents the degree of similarity and dissimilarity. For example, there are some data points clustered at the right area with yellow colour nodes showing that these design patterns in these regions having relatively similar characteristics according to some attributes of the dataset such as all sharing similar types of TLAs. There are also ranges of patterns grouped into clusters next to each other at the bottom middle of the map as can be seen, which is also a sign of similar patterns with similar characteristics found in these clusters. One of the common features found to be in these patterns is the frequent similarity in the learning objectives and the containment of similar interactive TLAs like questioning, dissuasion, and group activities. To better understand the feature similarities between these cluster nodes, Table [6.13] details some examples of design patterns with their most common features found. The table includes cluster ids 58, 68, and 69 as an example for the right region of the map which is represented in yellow. It also contains cluster ids 11, 19, 28, and 49 as an example of some of the group located at the bottom middle of the map witch is represented with dark blue colours.

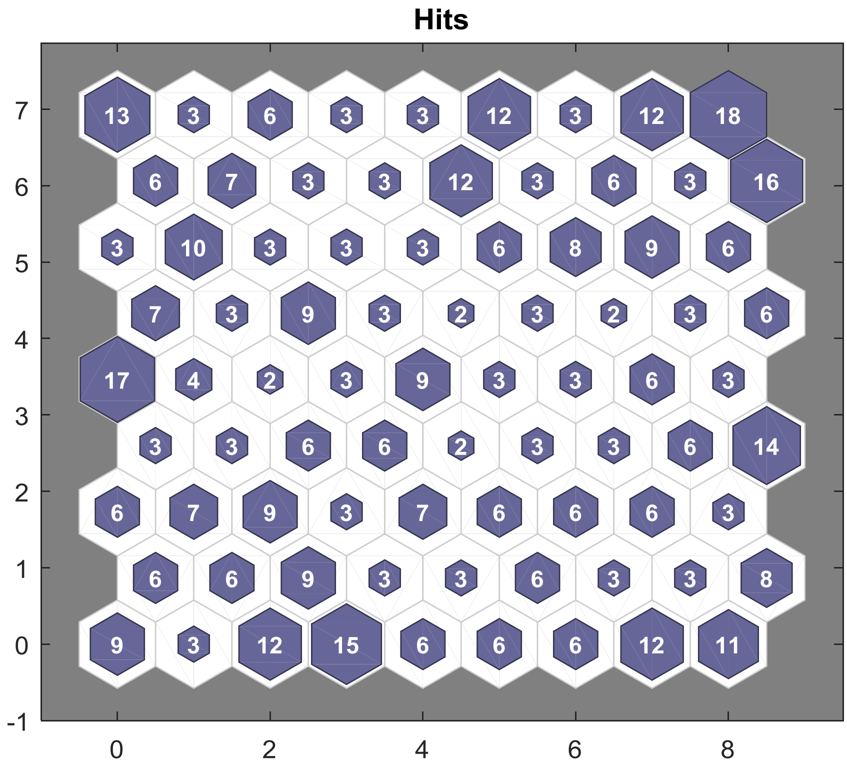


Figure [6.6]: SOM sample hits topology

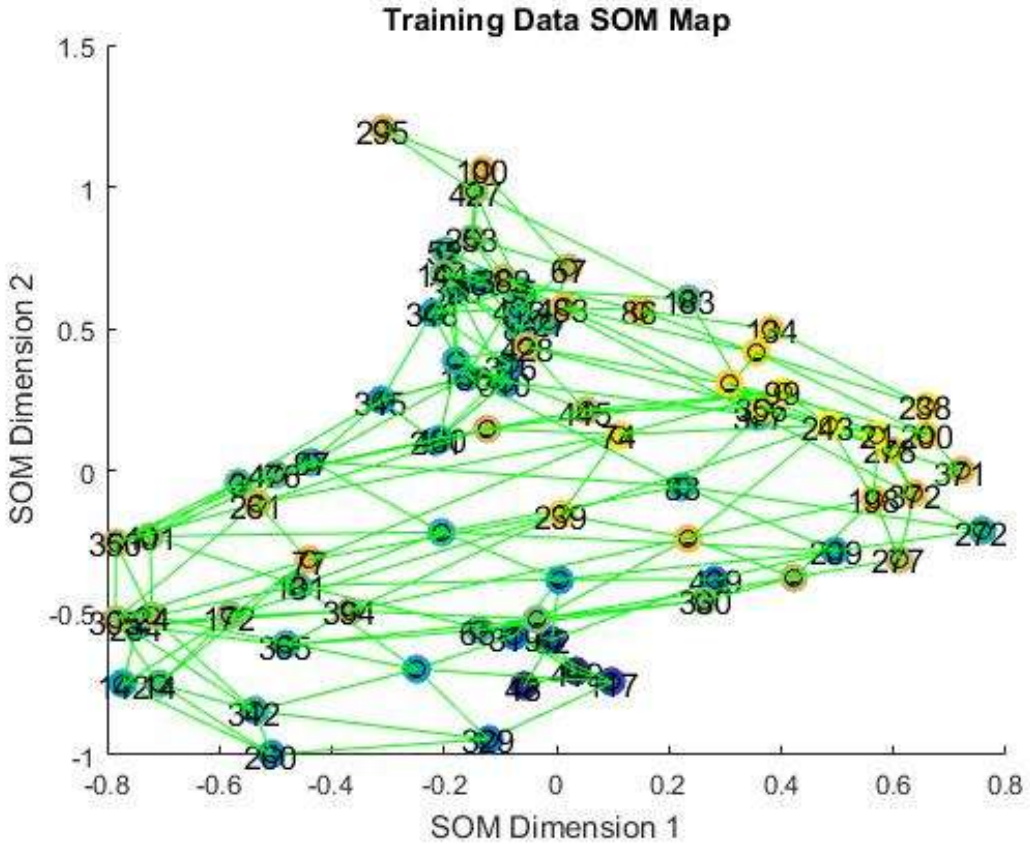


Figure [6.7]: SOM for the training patterns where a cluster node is labelled with the most representative design pattern i.e. the closest training pattern, which has the smallest Euclidean distance to the best matching unit (BMU) of that cluster node.

Table [6.13]: Comparison between some cluster nodes

Cluster id	Representative pattern id	Num. of Associated patterns	Features			
			LO	**LObjs**	**TLAs**	**ATs**
58	243	3	Applying (40%), Analysing (20%), Synthesis (40%)	Understanding (10%), Applying (40%), Evaluating (20%), Creating (10%)	Lecturers, Class discussion, Individual activities, Seminar exercises	Exam, Individual practical activity, Individual report
68	21	4	Applying (40%), Analysing (40%), Synthesis (20%)	Understanding (20%), Applying (30%), Analysing (30%), Evaluating (20%)	Lecturers, Individual activities, Seminar exercises	Exam, Individual practical activity, Individual report
69	278	3	Understanding (20%), Applying (40%), Analysing (20%), Synthesis (20%)	Applying (40%), Analysing (20%), Evaluating (20%) Creating (20%)	Lecturers, Class discussion, Individual activities, Seminar exercises	Exam, Individual practical activity, Individual report
11	319	3	Understanding	Understanding	Lectures, Seminars, Questioning, Group	Exam, Group practical, Practical

			(20%), Applying (40%), Creating (40%)	(20%), Applying (40%), Analysing (10%), Creating (30%)	discussion, Group practical activities,	report
19	117	8	Understanding (20%), Applying (60%), Creating (40%)	Understanding (20%), Applying (30%), Analysing (30%), Creating (20%)	Lectures, Seminars, Questioning, Group discussion, Group practical activities,	Exam, Group practical, Practical report
49	412	6	Applying (60%), Analysing (20%), Creating (20%)	Understanding (20%), Applying (40%), Analysing (10%), Creating (30%)	Lectures, Seminars, Questioning, Group discussion, Group practical activities,	Exam, Group practical, Practical report

After training the SOM with the training dataset, the evaluation was carried out on the testing dataset before pushing through the auto-encoder network and after using the obtained 9x9 map size. Figures [6.8] and [6.9] show the sammon's mapping plots for test patterns before and after transformation by the auto-encoder network. In each figure, each SOM node represents the BMU of the trained SOM activated by the test pattern. If a test pattern does *not* activate a SOM node that has any training patterns associated with it, it is not visually shown. The key difference between figure [6.8] (before test pattern transformation) and figure [6.9] (after test pattern transformation), is that more test patterns are associated with BMU of the trained SOM after transformation (figure [6.9]) and therefore test patterns have been nudged more closely to the good design patterns found in the training set. A common transformation that was made from figure [6.8] to figure [6.9] was that some raw test patterns weakly activated a BMU associated with a training pattern and after transformation was modified to more strongly match an alternative cluster of training patterns and therefore strongly activate a different BMU in figure [6.9]. For example, test pattern 19 was initially associated with a training BMU with a distance of 9.40 and after transformation, it was moved to a different BMU with closer distance of 7.12 as shown in figure [6.9]. In addition, some test patterns did not move from the initial BMU identified before pattern transformation, however, the transformed test pattern resulted in it being moved closer to the initial BMU assigned to it and thus the training patterns it represents. For example, test pattern 10 was initially assigned to a training BMU with a distance of 8.17 and after transformation, it was assigned the same BMU but with a shorter distance of 6.04 (as indicated in the tracking table [6.14]). The aim of this visualization is to track the best matching unite (BMU) locations for the most representative patterns displayed on the map and to visualize their movement. Further examples, can be seen in Table [6.14]. The table records the visualization trace and movement for those patterns with their distance from the cluster BMU, which is giving in bold. The table only lists those test patterns, before and after transformation that activated a BMU of the SOM that had training patterns associated with it. The other test patterns were considered 'too far' from a BMU representing one or more training patterns.

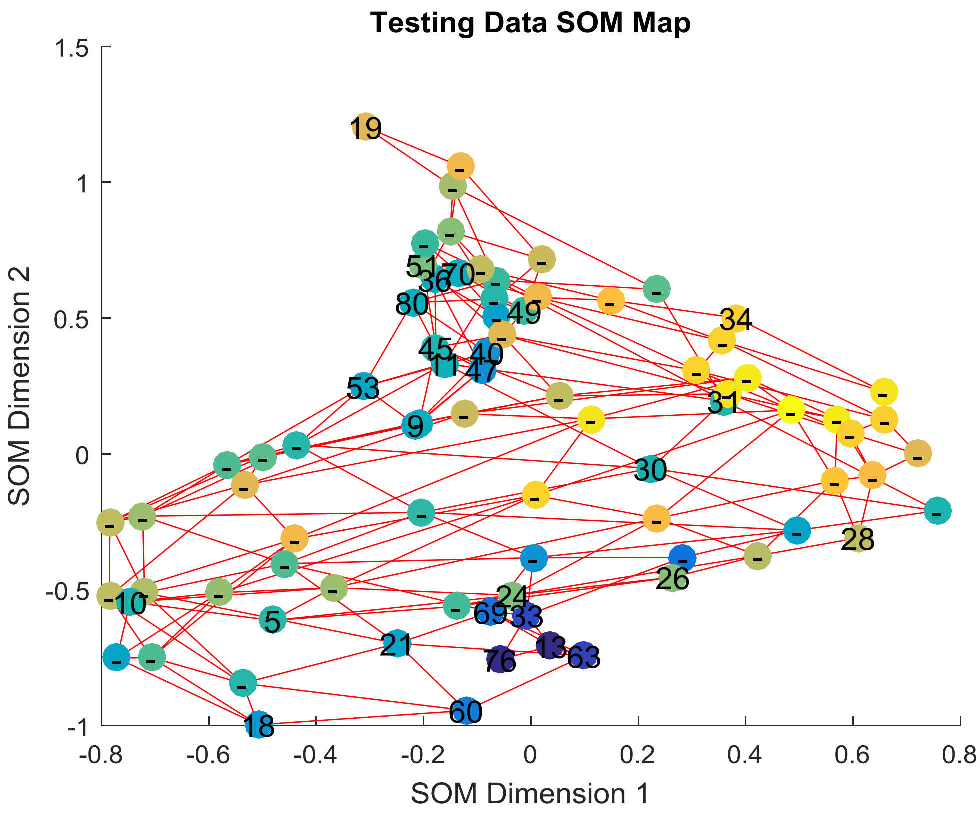


Figure [6.8]: SOM for the test patterns before applying auto-encoder

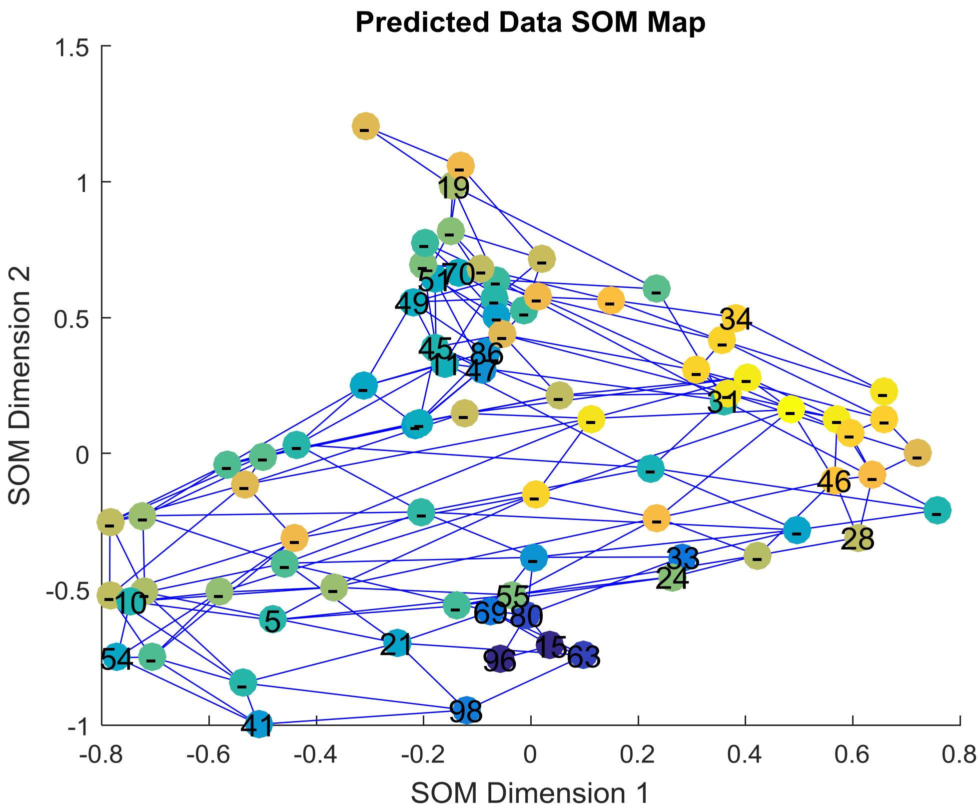


Figure [6.9]: SOM for the test patterns after applying auto-encode

Table [6.14]: Tracking table of most representative test patterns before and after transformation by the auto-encoder network with their distance from the BMU of the training nodes. The values are given correct to 2 decimal places as appropriate.

	BEFORE			AFTER	
TestPattern_id	Cluster_id	BMU	TestPattern_id	Cluster_id	BMU
5	64	8.32	**5**	64	5.83
9	33	8.88	**10**	55	6.04
10	55	8.17	**11**	15	6.28
11	15	6.95	15	10	5.64
13	10	6.74	19	8	7.12
19	9	9.40	**21**	20	7.07
21	20	9.13	**24**	74	6.00
24	65	8.41	**28**	76	6.21
26	74	7.94	**31**	6	6.81
28	76	7.97	**33**	3	6.42
30	14	9.24	**34**	81	6.81
31	6	8.31	41	37	6.78
33	2	7.36	**45**	52	7.12
34	81	9.41	46	67	6.78
36	44	8.25	**47**	43	6.74
44	64	9.62	49	25	5.51
45	52	10.02	**51**	44	6.04
47	43	8.60	54	46	7.56
49	36	6.64	55	65	6.36
51	53	9.08	**63**	19	6.05
53	24	11.00	**69**	11	5.89
60	28	7.77	**70**	45	5.67
63	19	6.58	**80**	2	6.10
69	11	6.79	86	34	6.64
70	45	6.55	96	1	5.21
76	1	5.87	98	28	7.08
80	25	7.65			

To conclude, the investigation provides an analysis and interpretation of the V1, V2, V3 and S before and after using a number of statistical and data visualization methods including the use of SOM. The purpose of this investigation was not having a sample for statistical purposes, but qualitative understandings regarding which main changes in module design elicit a high level student satisfaction and how strong these changes will impact or affect student satisfaction. From this Preliminary investigation, some core underlying design preferences were emerged as will be discussed in the next section.

6.3 Extracting the Core Design Principles Formed by the Neural Network

The network was trained on good design practices according to high student satisfaction. Then it was presented with input design patters associated with low satisfaction scores where its main task to recognize each of the given input pattern either as one it has seen before and therefore match it with a module within the training set or as one it might have seen before thus the new pattern is produced with such changes. These changes are interpreted as corrections the network attempts to make in order to match it with a good design from what it has learnt. It was seen in the analysis section above that there were significant differences between the alignment scores before and after especially in the V1 and V2 alignment scores. In this section these changes and the core learning design principles drawn from the network, to form such educational designs that can attract and raise the level of student satisfaction, are discussed and presented. The changes made to each case of V1, V2, and V3 will be discussed separately and conclude with the design principles extracted to effect improved student satisfaction in each case.

Case V1:

The analysis results of V1 alignment scores indicated that the network generated some changes that resulted in lowering the alignment values in most of the cases. This decrease in the degree of alignments has taken two forms where either the learning outcome's level has changed or the learning objective's level has changed. In case where the learning outcomes have changed; the network had made a weak change by moving down one level of the Bloom's level for the following learning outcomes: the analysis (Bloom 4), synthesis (Bloom 5), and evaluation (Bloom 6). This changed to application (Bloom 3), analysis (Bloom 4), and, synthesis (Bloom 5) respectively. These changes however, nudged the alignment scores of the test patterns (before) to become within the allowable range i.e. it has enhanced the relation from eliciting high alignment values above the range of 11.5. The network's decision in lowering the levels of the learning outcomes in general and focusing on suggesting outcomes with verbs under the application, analysis, and synthesis seems a good sign of its understanding in the relation of the learning outcome Bloom's levels. These learning levels can help students to acquire the required knowledge, skills, and ability that can help them to develop their higher learning abilities and this is exactly what Bloom's learning theory is based on where the assumption is that the six learning levels are progressive and that movement to higher levels depends on successful ability in the lower levels (Bloom, 1956). Therefore, focusing on the application, analysis, and synthesis more preferable and can lead to higher-learning improvement as well as higher student engagement and thus satisfaction. In case where the level of the objective has changed, it was noticed that it has been raised up one level in most cases, which can be seen as a strong change the network has made. This change (moving up one level of Bloom's for the given learning objective) can some time cause the alignment value to be positively misaligned in that learning objectives are eliciting higher Bloom's level than the associated learning outcome level. On the other hand, it was also noticed that the network has decreased the Bloom's level for some learning objectives (moving down one level of Bloom's for the given learning objective) resulting in negative misalignment between the learning outcomes and the learning objectives. Generally it can be said that the network is relatively ambivalent towards the alignment of learning outcomes and learning objectives suggesting there is some confusion between teaching practitioners as to how these are related.

Design preference 1: Both negative and positive misalignment is supported when associating learning outcomes and objectives – indicating some confusion in practice as to how these are associated and distinguished

Case V2:

The analysis results of V2 showed that there is an increase in V2 alignment scores resulting in an increase the satisfaction scores as well. Moreover, the effect size indicated an increase of one and a half standard deviations on V2, and is typically associated with a strong change in the type of activities and the level of Bloom's associated with the activities. In other words, we can say that the network was typically able to generate strong changes to the V2 alignment scores where it has been moving the level of the TLA from lower Bloom level to a higher level increasing the level of the corresponding student satisfaction. The following preferences were merged.

Design preference 1: Encouraging collaborative learning and high challengeable TLAs with low, intermediate, and high learning objectives help to improve student satisfaction

It is well known that collaborative learning is one of the most substantial approaches for improving learning outcomes (Beck, Chizhik, and McElroy, 2005; Chase and Okie, 2000; Hbscher-Younger and Narayanan, 2003; Jonassen, Lee, Yang, and Laffey, 2005; Joseph and Payne, 2003; McDowell, Werner, Bullock, and Fernald, 2002). It combines the social learning with experiential learning or inquiry-based learning in the sense that students work together in pairs or small groups to discuss concepts, or find solutions to problems. The move from individual-based activities to group-based activities was the most suggestions proposed by the network. The network demonstrates a preference for associating group-based activity, problem solving to learning objective eliciting Bloom's level 3, 4, 5 and 6. The network also preferred learning objectives beginning with the words 'define' and 'list, and learning objectives beginning with the words 'classify, identify, and 'explain', that is learning objectives in Bloom's level 1 and Bloom's level 2 respectively, to be associated with higher levels of learning as well. These preferences may be against the principle of constructive alignment, however it seems that it is better suited in practice and student satisfaction as it has been seen that when these kinds of associations are taking place, the average level of satisfaction increases from 3.3 to 3.8.

Design preference 2: Introducing plenty of example illustrations and exercises during the lectures help to increase the students' satisfaction level

Lectures play an important role in teaching for transmitting the knowledge however; pure lectures sometime can be pointless and hard to understand. Students like good and interactive lectures, otherwise they prefer group-based activities and more active learning to motivate them (Race, 2001, 2005). A preferred learning activity informed by the network was the use of example illustrations and small group exercises for learning objective being one of the following: application, analysis, and synthesis. If lectures were used as TLAs with the above objectives, the network suggestion was to associate the lectures with either example illustrations or group exercises. Incorporating plenty of examples during the given lecture often enhances both the presentation of the material and students' learning, which can help them to learn by applying a concept to real life. This in return reflects the levels of their satisfaction.

Design preference 3: Increasing the level of communication and interaction has a clear impact on students' learning and satisfaction

Interaction among students and instructors and among student themselves highly correlates with the level of their learning, engaging, and satisfaction (Swan, 2001). Weimer (1990) identified different strategies and methods that can be employed in the classroom that help enhance the learning experience of students. According to her, warming up the environment is integral in enhancing student satisfaction. Questioning, hands-on, and class discussion are one of the activities that help to create an environment of warmth, respect, enjoyment, and enthusiasm. These activities also increase the level of communication among students. The frequency relationships analysis of V2 showed that the most frequent type of teaching activities associated with learning objectives sitting at Bloom's levels 1 and 2 are found to be higher level activities than the associated level of the learning objectives. These activities were in the form of group discussions, class discussions, and questionings which are inline with the other studies such as Hiltz (1994), Moore (1989), and Swan (2001) that signify the impact of these teaching activities on achieving high student satisfaction and point the importance of increasing opportunities for interaction and communication during the lecture session.

Case V3:

Designing the assessment task is often one of the difficult parts that face the module designer. It is often viewed as being somehow separate from the learning process that used to measure what students know and what they don't know. However, assessment is an integral part of the learning process and should aim to improve the quality of student learning (Ciara O'Farrell, 2009). Biggs (1999), (2003), asserts the relationship between the module learning outcomes and the kind of assessment tasks used by ensuring that the intended verb in the learning outcomes is present in the assessment task. However, module designers often use one type of assessment in the form of exams to assess a student's knowledge. The analysis result of V3 showed that there is decrease in the V3 alignment values in order to bring the alignment values within the allowable range, however; this decrease in the V3 values was most affected by the decrease of the Bloom's level of the learning outcomes as V3 calculates the relationship between the learning outcomes and the assessment tasks. Practice, on the other hand, has revealed a modification to the alignment theory by using assessment tasks higher than learning outcomes. It also revealed the use of essay exam as alternative assessment strategy can be more effective tool than the traditional unseen exams.

Design preference 1: higher levels of assessment tasks can challenge the learning of the students to make them motivated towards learning.

Students find it interesting to learn when they figure out that their learning is based on challenging tasks. This helps them to develop their interest in the module and find themselves to be more challenged towards testing their learning and finding out what they have learned. With more challenges, they receive more motivation, and with more motivation, they find more level of satisfaction within the learning process and find it effective to be in a position where their learning is challenged at every step (Munns, 2009). Therefore, teaching practitioners are advised to use appropriate types of high-level assessment tasks. The transformed design patterns have showed that group assignments and group projects were associated to an average of 3.9 of the overall satisfaction.

Design preference 2: Essay exams can be more effective assessment tool than traditional unseen exams.

This gives an inference that using traditional exams to assess learning outcomes may not be always good practice and teaching practitioners may need to consider differ approaches to assess outcomes more effectively. Essay exams measures higher-order learning (Biggs, 2003), and also can be in perfect alignment relationship with all learning outcomes (knowledge, comprehension, application, analysis, synthesis, and evaluation) if teaching practitioners can make sure that the essay questions in the exam are aligned with their intended learning outcomes through activating the verb in the questions. This type of assessment has some advantages compared to the unseen vague examinations that make students revise the whole materials and test their memories rather than their understanding. Practice also revealed that there is good correlation (r = 0.415, significant at the 0.05 level) between high assessment tasks (Bloom's 6), in which an essay exam is categorised under this level, and student satisfaction. Accordingly Essay examinations are preferred over traditional unseen examinations for improving alignment and student satisfaction. The research literature appears to support this as essay type examinations require less memory and gives a better (compared to non-essay examinations) evaluation of how students have understood the subject and their ability to apply their knowledge and understanding (Brown, 2001; Champlin, 2006; Murphy, 2009; O'Farrell, 2009). Thus essay examinations can be considered a more effective assessment tool compared to the unseen examinations that make students revise the whole material and test their memories rather than their understanding. Whereas in real practice, students do need to be assessed on their critical thinking, understanding, communication, and collaboration skills that will help them in their future and career.

6.4 Summary

The given results emphasized two important factors, first the understanding of student satisfaction toward the core educational design components and identifying which of the components have significant roles in increasing the satisfaction. Second, searching and highlighting the most effective design preferences that teaching practitioners can utilize to improve both module alignment and student satisfaction. The outputs of the auto-encoder neural network creates prevalent patterns that show that most of the learning outcomes were associated with learning objectives formulated at the intermediate levels of Bloom's taxonomy of cognitive demand. This does not negate the importance of the other Bloom's cognitive levels, but explains that the intermediate levels of Bloom's taxonomy (application and analysis) were better appropriate in practice. Different teaching strategies utilizing higher level activities, which can be applied to different and similar modules, were presented and showed to correlate positively with student satisfaction. These results extracted from the neural network were consistent with the results of other studies like Killen (2009) and Kennedy et al (2007) which suggest that learning should be verbalized with a focus on intermediate and higher-order cognitive skills, and the learning activities should challenge students to make the best use of their learning experiences. The next chapter compares the system's performance with respect to other approaches and discusses the nexus of theory and practice. Then it summarises the whole research and draws its final conclusions and recommendations for future work.

CHAPTER 7: Conclusion and Future Work

7.1 The Nexus of Theory and Practice

In examining the relationship between learning theory and instructional design it is well known that the two disciplines affect each other continuously, each influencing the other for the good of the system and the good of the students (Desmarais, 2009). The theoretical framework is an important factor and considered the main building block for the design of a successful learning. In addition, the importance of linking between theory and practice in the design and development of any education system is also an important factor that helps to motivate the learning at the highest level. The task of translating a learning theory such as the 'Constructive Alignment' into practical application one of the difficult tasks (Ertmer and Newby, 2013), however; its application shows its effectiveness and its reflection on increasing the level of student satisfaction and perceived engagement in their learning, indicating the benefits of the application of this outcome-based pedagogical theory.

7.2 Comparison with Previous Work

Existing learning design tools, as discussed in Chapter 2, suffer from lacking a quantitative measure on which to base alignment or any means in which to adapt design patterns according to the student experience. Therefore, the development of EDIT aimed to introduce a neurally-inspired approach to learning design tools that seek to address this gap by implementing Tepper's quantitative measure of constructive alignment and associating module design patterns with their student satisfaction levels to calibrate alignment measures and provide more pragmatic and realistic design decisions based on both theory and practice. In this section a brief comparative review is presented between the existing learning design tools and EDIT, which is given in Table [7.1] providing the main properties.

Table [7.1]: Comparative review of main properties of EDIT and existing design tools

Learning design tool	Properties				
	Design level	Measure design quality?	Inference?	Adaptive?	Recommendations to enhance design?
LAMS	Session level (sequencing activities within a session)	No	No	No	No
Phoebe	Module and session level	No	No	No	No – just wiki-based references links and resources
LPP	Module and session level	No	No	No	No – only maps different design components together
LDSE	Module and session level	No	Yes inferences only from theory informed knowledge-based	No based on symbolic rules stored in the KB	Yes - recommends only alternative TLAs on the basis of the properties of the currently used TLA and suggests ways to combine TLAs and TEL approaches
EDIT	Module and session level	Yes provides numerical measure of alignment between module's components and for the entire module	Yes inferences from theory and real-practice informed knowledge-based	Yes based on new design patterns and adaptive KB	Yes - recommends alternative LOs, LObjs, TLAs, and ATs to effect better alignment and increase student satisfaction

Most learning design tools combine functionality for designing and supporting teaching practitioners through the series of design decisions involved in bringing together into a learning design the core elements of the education design together with advice and guidance on making those decisions. However, recommendation to enhance the design was given in the form of wiki-based such as in Phoebe or more test-based derived from a static knowledge base which has been based solely on theoretical frameworks such as LAMS and LPP. The most comparable tool is that of Laurillard (2011) who produced the LDSE investigating the approach of artificial intelligence by drawing inferences from comparisons between a user's decisions and a developing knowledge based system of design practice informed by pedagogic theories. A set of self-configurable rules are used as means to extend the knowledge base inference which provides a knowledge-aware application in finding, using and presenting the support to the user. This enhances the user's experience by offering alternative TLAs only for the given learning outcome to maximise the learner's potential to meet the learning outcome. However, a limitation of LDSE is again the interaction with a theory-informed knowledge base in drawing such inferences.

In contrast, *EDIT* has incorporated high scores of student satisfaction to indicate good design practices in the knowledge base and uses a variation of a back-propagation technique 'auto-associative neural network' that is trained to learn the relationships between good module designs to form a learned knowledge based system that can be used to 'correct' poor design patterns and build its inferences based on its knowledge base of theory and practice. The network has units associated with input and output as well as a modified set of hidden units that enable the network to learn a useful representation of the well-formed design patterns. With 519 training patterns generated from good practices according to high satisfaction scores, being presented to the auto-encoder network to learn from, the network was able to recognize 456 design patterns on training data correctly producing an average successful learning of 88% but failed to learn from the remaining 11 % that is about 57 design patterns. EDIT's ability to generalize from what it has learned to new patterns indicates that some general knowledge of effective design preferences (in line with pedagogic theory like the collaborative based learning and social-constructivist learning) has been extracted from its experience of good module designs. After successful training, 102 new test patterns associated with low student satisfactions were presented to the network where only 5 design patterns were recognized and 97 new design patterns were generated. The high RMSE generated from the test stage is a clear indication of aggregate design changes gained by the networks. The results of the analysis and interpretation of these changes revealed a set of design preferences inline with the social-constructivist learning and collaboration learning, which emphasize the discussion and collaborative nature of much learning (Vygotsky, 1978; Swan, 2001). The network also revealed its ability to make some modifications to alignment theory by suggesting higher level TLAs than their associated learning objectives and higher ATs than their associated learning outcomes as seen in Chapter 6.

In comparison with LDSE, *EDIT* provides more options for a given TLA which can be seen as alternative activities to consider beside the traditional lectures and seminars. In Table [7.2] it can be seen that *EDIT* provides an opportunity for more active and collaborative learning activities to be considered when the TLA 'lecture' was presented. LDSE on the other hand suggested only one TLA that intended to save the teacher's time. However, teacher-students interaction is one of the important factors that help students' engagement and contribute to their satisfaction (Swan, 2001). Moreover, providing alternative options for the given TLA can show a path to help the teacher or lecturer to pay more attention on the importance of supporting lectures and injecting them with these types of teaching activities due to their significant impact. The other examples in the table show that for each TLA in LDSE, the alternative activity is to use a technology-based activity that can aid again to save the teacher's time and cost while it does not consider if it suits practice or not. For example, considering a given TLA such as individual practical activity or resource-based individual activity, the LDSE's alternative suggestions to these TLAs are only: adaptive TEL individual activity, individual project activity, or TEL resourced based individual activity as seen in Table [7.2]. In case of EDIT for each of the individualised activities, the option of group activities is always suggested representing it as one of the best TLAs to use.

Table [7.2]: Example comparison between EDIT and LDSE in making alternative recommendations to the given TLAs

TLA	LDSE	EDIT
Individual practical activity	Adaptive TEL Individual activity, Individual project	Group activity, seminar excises, Individual project, laboratory notebook
Resource-based	TEL Resource-based Individual	Individual Project, Example illustration, group

Individual activity	activity	activity, TEL-Individual activity
TEL Resource-based Individual activity	Individual practical activity	Resource-based group activity, Group practical activity.
Teacher presentation	Online teacher presentation	Small group discussion, Class discussion, Group exercise, Example illustration,
Resource-based group activity	TEL Resource-based group activity	Resource-based group activity, Problem solving, Group presentation

Last but not least *EDIT* outperforms all the existing tools in its ability to recommend alternative types of objectives, activities, and assessment tasks and was not restricted to particular types as the case in all the tools in the table, which either they do not recommend at all or are limited in proposing only TLAs to enhance the design and unable to recommend alternative types of learning objectives and assessment tasks relative to the learning outcomes to effect better alignment.

7.2 Calibrating Alignment Ranges

Good and effective module design practices associated with high levels of student satisfaction scores were used to calibrate the alignment measures and identify meaningful alignment value ranges for the three main relations (V1, V2, and V3) for the alignment metric. Applying the metric to the module design patterns in the training set, which represents the good and effective module design practices, has resulted in the alignment value ranges shown in Table [7.3]. Therefore it is expected that if module designs stay within these ranges then the modules will be well-formed and constructively aligned in a way that will potentially yield positive student satisfaction.

Table 7.3: Acceptable and meaningful alignment value ranges calculated from good practise

	Min (X)	**Max (X)**
V1	4.8	11.1
V2	6.9	11.5
V3	7.8	13.1
AVG_V	7.6	10.7

The changes applied by the auto-encoder network to the test patterns indicated some design suggestions to bring the V1, V2, and V3 of the test patterns (before) into the allowable alignment ranges and move them closer towards the good design space as illustrated in Table [7.4] and therefore, raised the average satisfaction scores accordingly from an average of 3.3 to 3.8

Table [7.4]: Alignment ranges for test patterns before and after network's changes

Alignment Ranges	**V1**	**V2**	**V3**	**AVG_V**	**S**
Acceptable Alignment Ranges based on good practices (Training patterns)	4.8 – 11.1	6.9 – 11.5	7.8 -13.1	7.6 – 10.7	4.5
Test patterns (Before)	3.7 - 21.2	3.7 - 16.4	4.0 - 16.7	4.4 – 14.3	3.3
Test patterns (After)	4.8 - 14.8	5.8 - 15.1	3.8 – 16.6	5.7 – 13.3	3.8

In answering the question regarding the relationship between good module design and student satisfaction of the implementation and delivery of that design changes, the findings indicate that there is a relationship between the V1, V2, and V3 in relation to student satisfaction whereas the V2 alignment scores (r = 0.645, $p < 0.05$) was relatively highly correlated in comparison with V1 and V3 alignment scores (r = 0.504, $p < 0.05$) (r =0.415, $p < 0.05$). This correlation was confirmed with 41% of the variance in student satisfaction was accounted for the changes made to V2 eliciting a high level student satisfaction. The major conclusion that can be drawn from this correlation is to confirm that there is a strong link between the amount of active, social, and collaborative activities that student perceive and their satisfaction. Therefore, teaching practitioners are advised to increase these types of activities when designing their course and modules. The research shows that student satisfaction scores are good indicators to enhance module designs and learning. The results also show that neural networks offer a viable alternative to traditional artificial intelligence methods as a means of developing intelligent decision making tool, however; future works were identified for extending the scope of this research.

7.3 Summary of EDIT's Contributions

There currently exists no other learning design system that is able to objectively measure the quality of a learning design based on the principle of constructive alignment or any means in which to adapt design patterns according to their effectiveness in practice. This research is the first in quantifying the quality of learning designs by integrating both the principle of constructive alignment and good design practices based on threshold levels of student satisfaction. EDIT is an attempt to provide teaching practitioners with more pragmatic design solutions that is theoretically sound and aligned with current design practices within the discipline. Moreover, EDIT's transformation of test patterns can be used as future training patterns and thus new module designs. EDIT is an adaptive system so as practice evolves so too can EDIT's underlying knowledge-base by retraining the auto-encoder network. Furthermore, this research has led to the development of a substantial module design database with more than 500 design patterns for the science and technology sector provided in a structured way – so that relations between design components are easily understood and can thus be utilised by other researchers to evaluate their educational design tools and patterns.

Finally, in Tepper's alignment model there was no clear consensus as to what the threshold alignment values should be for V1, V2 and V3. In this research, effective practices (as judged by students satisfaction scores) have been used to calibrate these metrics and have thus identified suitable alignment ranges for each of the three tree structures as shown in Chapter 4, which researchers and practitioners can utilize and extend.

7.4 Conclusion and Future Work

Designing outcomes-based learning is a very complex and time-consuming process, yet fundamental to what teaching practitioners do. Different learning design tools exist but they do not measure the quality of an educational design, integrate theory and effective practice, or adapt to changing practices. This research thesis represents a significant step towards demonstrating the use of artificial neural networks for adaptively supporting the educational design process. More specifically, the approach uses an ‘auto-encoder neural network’ that is trained to memorise features of good module designs to form a learned knowledge based system that can then be used to ‘correct’ poor module designs during testing. EDIT’s ability to generalize from what it has learned to new patterns indicates that some general knowledge of effective design preferences (in line with pedagogic theory like the collaborative based learning and social-constructivist learning) has been extracted from its experience of good module designs. EDIT represents a data-orientated and objective view of design practices and is therefore dependent on the veracity of its input data. Further research will focus on generating larger samples of module designs that reflect practices across broader subject disciplines and higher education institutions. This will involve the use of focus groups that will also help to better establish practices surrounding the use of learning outcomes and learning objectives as there appears much confusion in this area. The research will also continue to investigate the design preferences of EDIT and how the transformations it makes actually works in practice i.e. do the projected improvements in student satisfaction actually materialise? Finally, further investigation is required as to how to optimise the deep auto-encoder networks so that they are able to act as perfect memories of the good design patterns as, at present, the current level of 88% indicates substantial scope for improvement.

References

Alexander, C. (1977). A Pattern Language: Towns - Buildings - Construction. Oxford University Press.

Almerico, G.M. and Baker, R.K. (2004). "Bloom's taxonomy illustrative verbs: developing a comprehensive list for educator use". Florida Association of Teacher Educators Journal, 1(4), pp. 1-10.

Anderson, L. and Krathwohl, D. A (2001). Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives New York.

Arbaugh, J B (2014). " System, scholar, or students? Which most influences online MBA course effectiveness?" Journal of Computer Assisted Learning, 30(4), 349-362.

Atkinson, S. (2011). "Embodied and Embedded Theory in Practice: The Student-Owned Learning-Engagement Model". London School of Economics and Political Science, UK.

Bação, F., Lobo, V., and Painho, M. (2008). "Applications of Different SelfOrganizing Map Variants to Geographical Information Science Problems", In Agarwal, P., and Skupin. A., editors, Self-Organising Maps: Applications in Geographic Information Science, pp.21–44, Wiley

Bailey D., and Thompson, D. (1990). "Developing neural-network applications", Published in Journal A.I. Expert Volume 5 Issue 9, Pages 34 – 41, ISSN:0888-3785

Battiti, R., and Masulli F. (1990). "BFGS Optimization for Faster and Automated Supervised Learning", INCC 90 Paris, International Neural Network Conference, pp 757–760.

BC College and Institute Student Outcomes (2003). "Understanding Student Satisfaction".[Online]:https://www.kpu.ca/sites/default/files/downloads/Understanding_Student_Outcomes6359.pdf

Beetham, H. and Sharpe, R. (2007). Rethinking Pedagogy for the Digital Age. Designing and delivering e-learning. Routledge: London.

Beesey, C., Davie, L., and Savin, D. (1991). Collecting Data. South Melbourne, Vic.: Macmillan.

Bengio, Y. (2007). "Learning deep architectures for AI," Dept. IRO, University de Montreal, Tech. Rep. 1312.

Bengio, Y. (2009). "Learning Deep Architectures for AI". FNT in Machine Learning Foundations and Trends in Machine Learning, 2(1), 1-127.

Bengio, Y. (2012). Practical Recommendations for Gradient-Based Training of Deep Architectures Yoshua Bengio Version 2, Sept. 16th,2012.

Bergin, J. (2012). *Pedagogical Patterns.* [Online]:http://www.pedagogicalpatterns.org/.

Biggs, J. (1996). *Enhancing teaching through constructive alignment*. Higher education Volume (32) 347-364

Biggs, J. (1999). *Teaching for quality learning at university*. Buckingham: SRHE and Open University Press.

Biggs, J. (2002). Aligning the curriculum to promote good learning. In Proceedings of Constructive Alignment in Action: An Imaginative Constructive Alignment in Action, LTSN conference, Centre Point Conference Centre London.

Biggs, J. (2003). Teaching for quality learning at university. Buckingham: Open University Press/Society for Research into Higher Education. (Second edition)

Biggs, J.B. and Collis, K.E (1982). Evaluating the Quality of Learning: The SOLO Taxonomy. New York: Academic Press.

Biggs, J. and Tang, C. (2007). Teaching for Quality Learning at University Maidenhead: Open University Press/McGraw Hill.

Biggs, J and Tang, C. (2011). Teaching for Quality Learning at University, 4th Edition.

Bishop, C. M. (1995). “Neural Networks for Pattern Recognition”, Oxford University Press, Oxford.

Bloom, B. (1956). Taxonomy of Educational Objectives Handbook I: Cognitive Domain, New York: Longman.

Boulanger-Lewandowski, N. (2014). "Modeling High-Dimensional Audio Sequences with Recurrent Neural Networks", PhD research, University of Montréal.

Bourlard, H., and Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. Biol. Cybern. Biological Cybernetics, 59(4-5), 291-294.

Britain, S. (2004). ‘A review of learning design. Concept, specifications and tools’. JISC report.

Brown, G. (2001). “Assessment: A Guide for Lecturers”. [Online]: http://www.palatine.ac.uk/files/980.pdf

Cabada, R.Z., Estrada, M.L., and Garcia, C.A. (2011). “EDUCA: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network”.

Cameron, L. (2011). “Could pedagogical planners be a useful learning design tool for university lecturers?” International conference on information communications technologies in education, Rhodes, Greece.

Cavanagh, M. (2011), “Students’ experiences of active engagement through cooperative learning activities in lectures”. Active Learning in Higher Education 12(1): 23–33.

Chalmers, D. (2008). “Teaching and Learning Quality Indicators in Australian Universities”.

Chang, Kuan-Yu. (2013). “Factors Affecting Student Satisfaction in Different Learning Deliveries”. Illinois State University.

Charlton, P. and Magoulas, G. (2010). Autonomic Computing and Ontologies to Enable Context-aware Learning Design.

Charlton, P., Magoulas, G. and Laurillard, D. (2009). Designing for Learning with Theory and Practice in Mind. London Knowledge Lab, University of London, UK.

Cashin, W. E. (1990). Improving lectures. In Teaching college: College readings for the new instructor, edited by M. Weimer and R. A. Neff, 59-63. Madison, Wis.: Magna Publications.

Christopher, A., Ishikawa, S., and Silverstein, M. (1977). “A Pattern Language” Oxford, Oxford University Press.

Christos, S., and Dimitrios, S. (2015), “NEURAL NETWORKS**”** [Online]: https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

City University of Seattle. (2013). “2012-2013 Student Satisfaction Survey: Analysis of Results”, USA 2013.

Coffey, M. and Gibbs, G. (2001). “The Evaluation of the Student Evaluation of Educational Quality”.

Cohen, J. (1988). “Statistical power analysis for the behavioural sciences (2nd ed.)”, Lawrence Earlbaum Associate, Hillsdale, NJ

Cohen, S. A. (1987). “Instructional alignment: Searching for a magic bullet”. Educational Researcher, 16(8): 16-20.

CompendiumLD: CompendiumLD Learning Design Software, **(2012)**. [OLnline]: http://compendiumld.open.ac.uk/

Conole, G. (2007). “Describing learning activities: tools and resources to guide practice in Rethinking pedagogy for a digital age” H. Beetham and R. Sharpe (Eds), Oxford.

Conole, G. (2008). “Using Compendium as a tool to support the design of learning activities”. In Okada, S. Buckingham Shum & T. Sherborne (Eds). Knowledge cartography – software tools and mapping techniques. London: Springer.

Conole, G. (2013). “Designing for learning in an open world”**.** New York, NY: Springer.

Cottrell, M. (1998). “Theoretical aspects of the SOM algorithm”. Neurocomputing**.**

Course Experience Questionnaire (CEQ) Report, (2009). [Online]: https://www.jcu.edu.au/__data/assets/pdf_file/0003/36291/jcu_133005.pdf

D'Andrea, V. (1999). "Organising Teaching and Learning: Outcomes-based Planning". In Fry, H., Ketteridge, S., and Marshall, S., (Eds) A Handbook for Teaching and Learning in Higher Education: Enhancing Academic Practice, London.

Dalziel, J. (2003). 'Implementing learning design: the Learning Activity Management System (LAMS)'. Proceedings of the ASCILITE 2003 conference

Dalziel, J. (2009). "Prospects for Learning Design research and LAMS". Teaching English with Technology, Special edition on LAMS and learning design.

Demuth, H., and Beale, M. (1998). "Neural network toolbox for use with MATLAB: User's guide". Natick, MA: MathWorks.

Desmarais, P. (2009). The Intersection of Educational Theory and the Practice of Instructional Design - Learning Theory and Educational Technology. [Online]: https://sites.google.com/a/boisestate.edu/edtech504/paul-desmarais

Duch W, Adamczak R, Grabczewski K. (1996). "Extraction of logical rules from training data using backpropagation networks". First Polish Conference on Theory and Applications of Artificial Intelligence.

Enzo Busseti, Osband, I., Scott, W .(2012). "Deep Learning for Time Series Modeling" CS 229 Final Project Report.

Erman Yukselturk and Zahide Yildirim. (2008). Journal of Educational Technology & Society,Vol. 11, No. 4 (October 2008), pp. 51-65.

Ertmer, P.A. and Newby, T.J., (2013). Behaviorism, Cognitivism, Constructivism: Comparing critical features from an instructional design perspective. Performance Improvement Quarterly, 26(2), pp.43-71

Firesmith, B. D. (2005). Use Cases: The Pros and Cons, [Online]: https://www.cs.hmc.edu/~mike/courses/mike121/readings/reqsModeling/firesmith.htm

Fletcher, R. (1975). "Practical Methods of Optimization", John Wiley & Sons.

Francis, W.N. and H. Kucera. (1979). Manual of information to accompany a Standard Sample of Present-day Edited American English, for use with digital computers. Department of Linguistics, Brown University.

G.E. Hinton and R.R. Salakhutdinov. (2006). "Reducing the Dimensionality of Data with Neural Networks", Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507.

Gershenson, C. Artificial Neural Networks for Beginners. [Online]: http://arxiv.org/ftp/cs/papers/0308/0308031.pdf

Glorot, X. and Bengio, Y.(2010). "Understanding the difficulty of training deep feedforward neural networks" Proc. AISTAT, 2010.

Goodfellow, I., Courville, A., and Bengio, Y. (2011). Spike-and-slab sparse coding for unsupervised feature discovery. In NIPS Workshop on Challenges in Learning Hierarchical Models.

Goodyear, P. (2005). "Educational design and networked learning: Patterns, pattern languages and design practice". Australasian Journal of Educational Technology, 21 (1), pp. 82--101.

Hanson, S. and J. Kegl. (1987). "PARSNIP", Proceedings of the Ninth Annual Conference of the Cognitive Science Society, Cognitive Science Society, 1987 Hillsdale, N.J

Henrad, F., (2010). "Learning our lesson: Review of Quality Teaching in Higher Education". [Online]: https://www.oecd.org/edu/imhe/44058352.pdf

H.H.Remmers and Plice, M. (1926). Reliability of rating at Purdue University, Industrial Psychology, Vol. 1 No. 11, November 1926.

Hinton, G.E. (1992). *How neural networks learn from experience, Scientific American*. Cite in *Rethinking Pedagogy for the Digital Age*, Beetham. H. and Sharpe, R. (Eds.). London: Routledge.

Hinton, G. E. and Zemel, R. S. (1995). Learning Population Codes by Minimizing Description Length. Neural Computation, 7(3), 549-564.

Hinton, G and Salakhutdinov, R. (2006). "Reducing the dimensionality of data with neural networks". Science, 313:504–507.

Hinton, G. E. (2010). A practical guide to training restricted Boltzmann machines. Technical Report UTML TR 2010-003, Department of Computer Science, University of Toronto.

Hornik, K. (1991). "Approximation Capabilities of Multilayer Feedforward Networks", Neural Networks, 4(2), 251–257.

Haykin, Simon (1998). "Neural Networks: A Comprehensive Foundation", Volume 2, Prentice Hall. ISBN 0-13-273350-1.

Jacobson, I. (1987). "Object-Oriented Development in an Industrial Environment", Proceedings of OOPSLA´87, SIGPLAN Notices, Vol. 22, No. 12, pages 183-191, 1987.

Jackson, N. (2000). "QAA: CHAMPION FOR CONSTRUCTIVE ALIGNMENT". LTSN Generic Centre and University of Surrey.

Jain Anil K., Jianchang, M., and Mohiuddin K.M.(1996). "Artificial Neural Networks: A Tutorial", IEEE Computational Science & Engineering Volume 29 Issue 3, March 1996 Page 31-44

JISC,The Design Studio (2013).The Design Studio/Institutional Approaches to Curriculum Design [Online]: http://jiscdesignstudio.pbworks.com/w/page/40489793/Institutional-Approaches-to-Curriculum-Design

JISC.(2006,2007). Design for learning. [Online]: http://www.jisc.ac.uk/whatwedo/programmes/elearningpedagogy/designlearn.aspx

Johansson, E.M., Dowla, F., Goodman, D. (1990). "Backpropagation Learning for Multi-Layer Feed-Forward Neural Networks Using the Conjugate Gradient Method", Lawrence Livermore National Laboratory, Preprint UCRL-JC-104850

Kaastra, I., and Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. Neurocomputing, 10(3), 215-236.

KATZ, J.O. (1992). Developing neural network forecasters for trading, Technical Analysis of Stocks and Commodities, 10 (4), 160-168.

Kazakhstan Institute of Management, Economics, and Strategic Research, (2011). Student Satisfaction Survey Report.

Keele University. (2012). "Quality Assurance General Guidance", U.K.

Kennedy, D., Hyland, A., and Ryan, N. (2007). "Writing and using learning outcomes: a practical guide". Date Views 05.11. 2013.

Kerschen J.C G.(2004). "Feature Extraction using Auto-associative neural network".

Killen, R. (2009). "Outcomes-based education: Principles and possibilities", University of Newcastle, Faculty of Education [Online]: http://drjj.uitm.edu.my/DRJJ/CONFERENCE/UPSI/OBEKillen.pdf

Kolb, D. A., and Fry, R. E. (1974). Toward an applied theory of experiential learning. MIT Alfred P. Sloan School of Management.

Kohonen, T. (1995). "Self organating maps". Berlin.

Kramer, M. A. (1991). "Nonlinear principal component analysis using autoassociative neural networks". AIChE Journal AIChE J., 37(2), 233-243.

Kramer M. A. (1992). “Nonlinear Principle components analysis using Auto-associative Neural Networks”, AlChE Journal.

Labhrain, Lain Mac et al. (2005). “Obtaining Student Feedback on Teaching and Course Quality”. CELT.

LAMS International (2009) LAMS, Learning activity management system. [Online] Available at http://www.lamsinternational.com

Laurillard, D. (2002). “Rethinking University Teaching: A Conversational Framework for the Effective Use of Learning Technologies” (2nd ed.). London. Routledge Falmer.

Laurillard, D. (2010). “An Approach to Curriculum Design. London Knowledge Lab”. Institute of Education.

Laurillard, D. (2012). “Teaching as a Design Science: Building Pedagogical Pattern for Learning and Technology”.

Laurillard, D., Charlton, P., Craft, B., Dimakopoulos, D., Ljubojevic, D., and Magoulas, G. (2011). A constructionist learning environment for teachers to model learning designs. Journal of Computer Assisted Learning.

Laurillard, D. and Ljubojevic, D. (2011). “Evaluating learning designs through the formal representation of pedagogical patterns In C”. Kohls & J. Wedekind (Eds.), Investigations of E-Learning Patterns: Context Factors, Problems and Solutions. Hershey, PA: IGI Global.

Laurillard, D. and Masterman, L. (2010). “A constructionist approach to collaboration through a learning design support environment: Balancing users’ requirements with researchers’ theory-informed aspirations”. Paper presented at the European LAMS and Learning. Design Conference, Wolfson College, Oxford University.

Lawrence J. (1991),”Data preparation for a neural network”.

LDSE: Learning Design Support Environment, (2011). [Online]: https://sites.google.com/a/lkl.ac.uk/ldse/

Levine, D. (2000). “Introduction to Neural and Cognitive Modeling”. Second edition.

Lo., Celia C.(2010), “How Student Satisfaction Factors Affect Perceived Learning”. Journal of the Scholarship of Teaching and Learning.

Lou M.(1993), “Preprocessing Data for Neural Networks”

LPP: London Pedagogical Planner, **(2007).** [Online]: http://www.jisc.ac.uk/publications/reports/2008/llpfinal.aspx

Li, N., Marsh, V. and Rienties, B., (2016). Modelling and Managing Learner Satisfaction: Use of Learner Feedback to Enhance Blended and Online Learning Experience. Decision Sciences Journal of Innovative Education,14(2), pp.216-242.

Machemer, P., and Crawford, P. (2007). Student perceptions of active learning in a large cross-disciplinary classroom. Active Learning in Higher Education 8(1): 9–30.

Mack, N., Woodsong, C. (2011). “Qualitative research methods: A data collector's field guide”. North Carolina: FLI.

Margery, D. (2003). *Outcome-Based Education* [Online]: http://www.utpjournals.com/jvme/tocs/303/258.pdf

Marton, F. and Saljo, R. (1976). On qualitative differences in learning I. Outcome and process. British Journal of Educational Psychology 46: 4-11.

McAndrew, P., Goodyear, P. and Dalziel, J. (2006). Patterns, designs and activities: unifying descriptions of learning structures. International Journal of Learning Technology, 2(2-3), pp. 216–242.

McNamara, C. (2008). Overview of methods to collect information. Minneapolis, MN: Free Management Library [Online]: http://www.managementhelp.org/evaluatn/fnl_eval.htm#anchor1585345

MELCOE. (2009). Macquarie E-learning Centre Of Excellence (MELCOE) - Current Projects Available at: http://www.melcoe.mq.edu.au/current.htm

M. Ghonjia, Z. K. (2015). Analysis of the some effective teaching quality factors within faculty members of agricultural and natural resources colleges in Tehran University. Journal of the Saudi Society of Agricultural Sciences, 14(2), 109–115.

Metcalfe, J. (1991). Recognition failure and the composite memory trace in CHARM. Psychological Review, 98, 529-553

Michael, N. (2016). Neural Networks and Deep Learning online book. [Online] Available at http://neuralnetworksanddeeplearning.com/index.html

Militaru, Gheorghe et al.(2015) “An Exploratory Study of Student Satisfaction: The Moderating Role of Digital Technologies”.

Moller.(1993). “Neural Networks”, Vol. 6, 1993, pp. 525–533

Moore, M.G. and Kearsley, G. (1996). Distance education: A systems view. Belmont.

Muhammad Dilshad, H. M. (2010). Quality Indicators in Teacher Education Programmes. Pakistan Journal of Social Sciences (PJSS), 30(2), 401-411.

Munns, D. S. (2009). Citation for Outstanding Contributions to Student Learning

Negnevitsky, M. (2005). *Artificial Intelligence. A Guide to Intelligent Systems.* Second edition.

Nottingham Trent University. (2013). Quality Handbook Supplement EvaSys, [Online]: https://www4.ntu.ac.uk/adq/document_uploads/quality_handbook/148421.pdf

NSS, (2013). Thestudentsurvey.com. 2013. *The National Student Survey* [Online] http://www.thestudentsurvey.com/the_nss.html#.Uo9sJMTIbOM

O'Farrell, C. (2009). Enhancing Student Learning through Assessment. Available at learningandteaching.dit.ie/documents/assessment_toolkit_v41f.pdf

PHOEBE: Pedagogy Project (2006). [Online]: http://www.jisc.org.uk/publications/reports/2008/phoebefinalreport.aspx

Pollack, J. B. (1990). "Recursive distributed representations". Artificial Intelligence, 46(1-2), 77-105

Pressman, R. (2001). *Software Engineering: A Practitioner's Approach*. Boston: McGraw Hill.

QAA (2013). Quality Assurance Agency for Higher Education [Online]: http://www.qaa.ac.uk/benchmark/index.htm)

Questionnaire (SEEQ). in UK Higher Education. Assessment & Evaluation in Higher Education, 26, 89-93.

Race, P. (2001). The Lecturer's Toolkit: A Practical Guide to Learning, Teaching and Assessment. London: Kogan Page.

Race, P. (2010). Making Learning Happen: a guide for post- compulsory education (2nd edn). London: Sage Publications.

Race, P. (2005). Making learning happen: A guide for post-compulsory education. London: SAGE Publications.

Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. (2007). Efficient learning of sparse representations with an energy-based model. In NIPS'06.

Rienties, B., Toetenel, L., and Bryan, A. (2015). 'Scaling up' learning design: impact of learning design activities on LMS behavior and performance. Paper presented at the 5th Learning Analytics Knowledge conference, New York.

Roschelle, J. (1992). "Learning by Collaborating: Convergent Conceptual Change". Journal of the Learning Sciences, 2(3), 235-276.

Rumelhart, D.E., G.E. Hinton, R.J. Williams (1986). Learning Internal Representations by Error Propagation, in: Parallel Distributed Processing: Exploration in the

Microstructure of Cognition, Eds. D.E. Rumelhart, J.L. McClelland, MIT Press, Cambridge, MA., pp 318–362.

Ruth, Williams et al. (2004). "Collecting and Using Student Feedback". Centre for Higher Education Research and Information.

Samarasinghe, S. (2007). Neural networks for applied sciences and engineering: From fundamentals to complex pattern recognition. Boca Raton, FL Auerbach.

Saunders, M. N., Lewis, P., and Thornhill, A. (2003). Research methods for business students. Harlow, England: Prentice Hall.

Setiono, R.(2000). "Extracting M-of-N rules from trained neural networks". IEEE-NN. 2000;11 (2):512.

Shaw, I.F. (1999). Qualitative Evaluation. London. International Social Work

Sinclaire, K. (2017). "Student Satisfaction with Online Learning: Lessons from Organizational Behavior". Research in Higher Education Journal.

Solinas, Guliana.(2012). "What Really Affects Students Satisfaction: An Assessment of Quality Through a University-Wide Student Survey", Creative Education 2012. Vol.3

Statistics for Psychology. (2016). Retrieved September 20, 2016, [Online]: http://staff.bath.ac.uk/pssiw/stats2/page2/page14/page14.html

Suarman.(2015)."Teaching Quality and Students Satisfaction: The Intermediary Role of Relationship between Lecturers and the Students of the Higher Learning Institutes". Mediterranean Journal of Social Sciences.

Sutskever I., Martens J., Dahl G., and Hinton G.(2013). "On the importance of momentum and initialization in deep learning", 30th International Conference on Machine Learning.

Swan, M. (2005). Improving learning in mathematics: Challenges and strategies. Sheffield, England: Department of Education and Skills Standards Unity

Swan, K. (2001). Virtual interaction: Design factors affecting student satisfaction and perceived learning in asynchronous online courses. Distance Education, 22(2), 306-331.

Swain, M., Dash, S. and Mohapatra A. (2012). An Approach for IRIS Plant Classification Using Neural Network.

Swansea University. (2014). "Policy on Student Module and Program Feedback", U.K.

Tepper, J. (2006). *Measuring Constructive Alignment: An Alignment Metric to Guide Good Practice*. Nottingham Trent University.

Tepper, J. (2008). *From Theory to Automata: A Computational Model of Constructive Alignment*. Nottingham Trent University.

Turian J. Ratinov L. Bengio Y.(2010). Word Representations: A Simple and General Method for Semi-Supervised Learning. In: Proc. Association for Computational Linguistics.

Uka, Ana.(2014). "Student Satisfaction As Indicator Of Quality In Higher Education" Journal of Educational and Instructional Studies in the World.

University of Birmingham. (2012). "National Student Survey", [Online]: http://www.birmingham.ac.uk/schools/sport-exercise/teaching/teaching-quality.aspx

Vanfretti, Luigi Farrokhabadi, Mostafa.(2012) "Implementing constructive alignment theory in a power system analysis course using a consensus model", 6th IEEE International Conference on E-Learning in Industrial Electronics, ICELIE 2012, IEEE , 2012, 94-100 p.

Vesanto J. (1999). SOM-Based Visualization Methods. Intelligent Data Analysis

Voegtlin, T. and Dominey, P. F. (2005). Linear recursive distributed representations. Neural Networks, 18(7), 878-895

Vygotsky, L. (1978). Mind in society: The development of higher psychological processes. Cambridge: Harvard University Press.

Weber, E. U., and Murdock, B. B. (1989). Priming in a distributed memory system: Implications for models of implicit memory. In S. Lewandowsky, J. Dunn, & K. Kirsner (Eds.), Implicit memory: Theoretical issues (p. 87-89). Hillsdale, NJ: Lawrence Erlbaum Publishers.

Williams, L. (2006). *Testing Overview and Black-Box Testing Techniques.*

Williams, Kaylene C. and Caroline C. Williams. (2011). "Five key ingredients for improving student motivation." Research in Higher Education Journal

Yin, R. (1994), Case Study Research, Design and Methods, Sage Publications, Newbury Park.

Yu, K., Salakhutdinov, R., Lecun, Y., Hinton, G., and Bengio, Y. (2009). Workshop summary: Workshop on learning feature hierarchies. Proceedings of the 26th Annual International Conference on Machine Learning

Zerihun, Z., Beishuizen, J., and Os, W. (2012). Student learning experience as indicator of teaching quality. Educational Assessment, Evaluation and Accountability, 24(2), 99-111.

Appendixes

Appendix [A]: Xml Structure of LDSE Pattern


```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ldse version="3.0">
<design type="module">
<module elapsedTime="0" name="Process Modelling" noOfStudents="20" staffTimeAllocated="0">
<calendar end="1371117572999" start="1371117572999"/>
<intendedLearningTime>0</intendedLearningTime>
<staffInvolved/>
<learnerCharacteristics/>
<prerequisites/>
<summativeAssessment/>
<aims/>
<sessions size="3">
<session elapsedTime="0" learningLevel="1" learningTime="60" name="UseCase Modelling Lecture"
noOfStudents="20" scheduleEnd="1262307600000" scheduleStart="1262304000000" topic=""
typeID="TeacherSupportedClassFE">
<id>1377260543043</id>
<description/>
<note/>
<reflections/>
<student_feedback/>
<activities size="4">
<activity description="" duration="15" groupsize="20" name="Introduction to use cases" notes="Describe the
role of use cases" start="1262304000000">
<teachingmethod group=" TeacherSupportedClassFE " id="DefaultTutorPresentation" name="Teacher
presentation"/>
<resources/>
</activity>
<activity description="" duration="20" groupsize="5" name="Creating use case tables " notes="Explain the
process used to create use cases.&#10;" start="1262304900000">
<teachingmethod group=" TeacherSupportedClassFE " id="DefaultResourceBasedGroupActivity"
name="Resource based group activity"/>
<resources/>
</activity>
<activity description="" duration="15" groupsize="20" name="Purpose of Process Modelling" notes="Define
process modelling and describe its benefits." start="1262306100000">
<teachingmethod group=" TeacherSupportedClassFE " id="DefaultTutorPresentation" name="Teacher
presentation"/>
<resources/>
</activity>
<activity description="" duration="10" groupsize="1" name="Fragment Logical level 0 DFD"
notes="Demonstrate an understanding of the basic concepts and constructs of the DFDs."
start="1262307000000">
<teachingmethod group=" TeacherSupportedClassFE " id="DefaultResourceBasedIndividualActivity"
name="Resource based individual activity"/>
<resources/>
</activity>
</activities>
<aims/>
<outcomes>
```

```
<outcome verb="Analyse (Analysis)"><![CDATA[Analyse sysytems in a systematic and methodical
manner.]]></outcome>
</outcomes>
</session>
<session elapsedTime="0" learningLevel="1" learningTime="60" name="DFD Levelling and balancing"
noOfStudents="20" scheduleEnd="1262311200000" scheduleStart="1262307600000" topic=""
typeID="TeacherSupportedClassFE">
<id>1377260543049</id>
<description/>
<note/>
<reflections/>
<student_feedback/>
<activities size="5">
<activity description="" duration="15" groupsize="20" name="Introduction to decomposing DFDs to level1"
notes="Breakdown DFDs into lower level diagrams." start="1262304000000">
<teachingmethod group=" TeacherSupportedClassFE " id="DefaultTutorPresentation" name="Teacher
presentation"/>
<resources/>
</activity>
<activity description="" duration="10" groupsize="1" name="Create level 1 DFD" notes="Breakdown DFDs into
lower level diagrams." start="1262304900000">
<teachingmethod group=" TeacherSupportedClassFE " id="DefaultResourceBasedIndividualActivity"
name="Resource based individual activity"/>
<resources/>
</activity>
<activity description="" duration="10" groupsize="20" name="Balancing DFD Across levels " notes="Explain
DFD balancing." start="1262305500000">
<teachingmethod group=" TeacherSupportedClassFE " id="DefaultTutorPresentation" name="Teacher
presentation"/>
<resources/>
</activity>
<activity description="" duration="5" groupsize="5" name="Balance DFD for a given system " notes="Explain
DFD balancing." start="1262306100000">
<teachingmethod group=" TeacherSupportedClassFE " id="DefaultResourceBasedGroupActivity"
name="Resource based group activity"/>
<resources/>
</activity>
<activity description="" duration="20" groupsize="20" name="Explain Types and uses of DFDs" notes="Explain
the differences among four types of DFDs and the use of DFDs in system analysis and design."
start="1262306400000">
<teachingmethod group=" TeacherSupportedClassFE " id="DefaultTutorPresentation" name="Teacher
presentation"/>
<resources/>
</activity>
</activities>
<aims/>
<outcomes>
<outcome verb="Formulate (Comprehension)"><![CDATA[Formulate a set of balanced DFDs for a simple
information
systems.]]></outcome>
<outcome verb="Analyse (Analysis)"><![CDATA[Analyse sysytems in a systematic and methodical
manner.]]></outcome>
<outcome verb="Design (Synthesis)"><![CDATA[Design the processing logic for primitive DFD processes using a
logic modelling
technique.]]></outcome>
</outcomes>
</session>
```

<session elapsedTime="0" learningLevel="1" learningTime="180" name="Summative Assessment F E" noOfStudents="15" scheduleEnd="1262314800000" scheduleStart="1262304000000" topic="" typeID="SummativeAssessmentFE">
<id>1377260543051</id>
<description/>
<note/>
<reflections/>
<student_feedback/>
<activities size="2">
<activity description="" duration="180" groupsize="1" name="Exam" notes="" start="1262304000000">
<teachingmethod group=" SummativeAssessmentFE " id="DefaultExam" name="Exam"/>
<resources/>
</activity>
<activity description="" duration="30" groupsize="1" name="Project" notes="" start="1262314800000">
<teachingmethod group=" SummativeAssessmentFE " id="DefaultProjectReport" name="Project"/>
<resources/>
</activity>
</activities>
<aims/>
<outcomes/>
</session>
</sessions>
</module>
</design>
</ldse>

Appendix [B]: Alignment Tables

Teaching and learning activities and the type of learning they elicit. The level in Bloom's taxonomy assigned for each assessment task based on Biggs 2003

TLA	A form of learning	Bloom's Taxonomy (1-6)
1. Teacher-controlled (TLAt)		
lecture, set texts (TLAt1)	reception of selected content	2
think aloud (TLAt2)	demonstrate conceptual skills	3
questioning (TLAt3)	clarifying, seeking error	4
advance organizer (TLAt4)	structuring, preview	5
concept mapping (TLAt5)	structuring, overview	5
tutorial (TLAt6)	elaboration, clarification	2
laboratory (TLAt7)	procedures, application	4
excursion (TLAt8)	experiential knowledge, interest	2
seminar (TLAt9)	clarify, presentation skill	3
2. Peer-controlled (TLAp)		
various groups (TLAp1)	elaboration, problem-solving, metacognition	6
learning partners (TLAp2)	resolve differences, application	6
peer teaching (TLAp3)	depends whether teacher or taught	3
spontaneous collaboration (TLAp4)	breadth, self-insight	3
3. Self-controlled (TLAs)		
generic study skills (TLAs1)	basic self-management	5/6
content study skills (TLAs2)	information handling	5/6
metacognitive learning skills (TLAs3)	independence and self-monitoring	6

Assessment tasks and the types of the type of learning they evaluate. The level in Bloom's taxonomy assigned for each assessment task based on Biggs 2003.

AT	**Type of learning assessed**	**Bloom's level. (1-6)**
1. Extended prose, essay-type (ATe)		
essay exam (ATe1)	rote, question spotting, speed structuring	5
open book (Ate2)	as above but less memory and greater coverage	2
assignment, take-home (ATe3)	read widely, interrelate, organise, apply, copy	5
2. Objective test (ATo)		
multiple-choice (ATo1)	recognition, strategy, comprehension, coverage	2
ordered outcome (ATo2)	hierarchies of understanding	3
3. Performance assessment (ATp)		
practicum (ATp1)	skills needed in real life, procedural knowledge	4
seminar, presentation (ATp2)	communication skills	3
posters (ATp3)	Concentrating on relevance, application	3
interviewing (ATp4)	responding interactively, recall, application	3
critical incidents (ATp5)	reflection, application, sense of relevance	6
project (ATp6)	application, research, problem solving	4
reflective journal (ATp7)	reflection, application, sense of relevance	6
case study, problems (ATp8)	application, professional skills	3
portfolio (ATp9)	reflection, creativity, unintended outcomes	6
4. Rapid ATs (large class) (ATr)		
concept maps (ATr1)	coverage, relationships, some holistic understanding	5
Venn diagrams (ATr2)	Relationships	2
three-minute essay (ATr3)	level of understanding, sense of relevance	3
gobbets (ATr4)	realising importance of significant detail, some multistructural understanding across topics	2
short answer (ATr5)	recall units of information, coverage	2
letter to a friend (ATr6)	holistic understanding, application, reflection	3
cloze (ATr7)	Comprehension of main ideas	2

Appendix [C]: Student Evaluation Survey

EvaSys | SST Undergraduate Module Evaluation Survey 12/13 | Electric Paper

MODULE NAME: ____________________ TUTOR NAME: ____________

MODULE CODE: ____________________

GROUP NO: ______

NOTTINGHAM TRENT UNIVERSITY

PLEASE READ BEFORE COMPLETING THIS SURVEY: This survey requests your feedback on this module. Your responses will be used to identify areas of good practice and improve the teaching and learning experience for you and your peers. Please mark only one box for each question.

1. Feedback on <u>group</u> based teaching e.g. Seminars / Labs / Practicals

		Definitely Agree	Mostly Agree	Neither Agree or Disagree	Mostly Disagree	Definitely Disagree	N/A
1.1	My tutor is well organised	☐	☐	☐	☐	☐	☐
1.2	My tutor is supportive	☐	☐	☐	☐	☐	☐
1.3	My tutor communicates clearly	☐	☐	☐	☐	☐	☐
1.4	My tutor is good at explaining things	☐	☐	☐	☐	☐	☐
1.5	I feel able to ask questions	☐	☐	☐	☐	☐	☐
1.6	I find these classes to be valuable learning experiences	☐	☐	☐	☐	☐	☐

1.7 Please use the space below to make specific comments on aspects of the teaching that helped facilitate your learning and/ or aspects that could be improved.

2. Feedback on module teaching (e.g. lectures, workshops)

2.1	The lectures are well structured and organised	☐	☐	☐	☐	☐	☐
2.2	For me, module teaching staff are enthusiastic about what they are teaching	☐	☐	☐	☐	☐	☐
2.3	I found the module teaching staff are good at explaining things	☐	☐	☐	☐	☐	☐
2.4	Information presented on teaching materials (including slides) is clear	☐	☐	☐	☐	☐	☐

2.5 Please use the space below to feedback to your module teaching staff. Please state lecturer name(s) followed by specific comments on aspects which were particularly helpful in facilitating your learning or those which could be improved.

Please turn over.

F78U0P1PL0V0 28.10.2012, Page 1/2

Electric Paper

3. Module Organisation and Resources

		Definitely Agree	Mostly Agree	Neither Agree or Disagree	Mostly Disagree	Definitely Disagree
3.1	The aims and outcomes of the module were explained clearly to me	☐	☐	☐	☐	☐
3.2	I found that the module focused on what was set out in the module document	☐	☐	☐	☐	☐
3.3	The assessment criteria were explained clearly to me	☐	☐	☐	☐	☐
3.4	I receive regular feedback on my progress	☐	☐	☐	☐	☐
3.5	The NOW online resources for this module help support my learning	☐	☐	☐	☐	☐
3.6	I have found the module to be well organised	☐	☐	☐	☐	☐

4. Overall Satisfaction

		Definitely Agree	Mostly Agree	Neither Agree or Disagree	Mostly Disagree	Definitely Disagree
4.1	I found the module intellectually stimulating	☐	☐	☐	☐	☐
4.2	I am satisfied with the quality of classroom and specialist teaching facilities used on the module	☐	☐	☐	☐	☐
4.3	Overall, I am satisfied with the **quality** of the module	☐	☐	☐	☐	☐
4.4	Overall, I am satisfied with the **teaching** on this module	☐	☐	☐	☐	☐

4.5 Aspects I have liked about the module:

4.6 Aspects I think should be changed about the module, and why. Please try to be specific:

5. Feedback and your Learning

		Definitely Agree	Mostly Agree	Neither Agree or Disagree	Mostly Disagree	Definitely Disagree
5.1	The feedback I have received on my work has supported my learning	☐	☐	☐	☐	☐

6. Student Engagement

6.1 Which of the following best describes your attendance in this module so far:

☐ 0-20% ☐ 21-40% ☐ 41-60%
☐ 61-80% ☐ 81-100%

6.2 Why do you think your attendance is good or poor in this module?

6.3 Three things I did that helped me learn:

Thank you for completing this Module Survey.

F78U0P2PL0V0 29.10.2012, Page 2/2

Appendix [D]: Mappings among the TLAs and ATs in SST and LDSE

TLAs

TLAs LDSE	TLAs SST
Session: Tutor-supported class	Session: Lectures
Teacher presentation	Lecture, presentation
Student presentation	Student presentation
Class discussion	Class discussion
Small group discussion	Small group discussion
Resource-based group/individual activity	group/individual activity
Group/individual practical activity	Group/individual practical activity, Group/individual Laboratory
Check for learning	Questioning
Online teacher presentation	Online lecture, virtual learning environment (NOW)
Session: Tutor-supported group	Session: Seminars/Labs
TLAs LDSE	TLAs SST
Teacher presentation	Tutorial, seminar
Student presentation	Student presentation
Small group discussion	Small group discussion
Resource-based group/individual activity	group/individual activity, problem solving,
Group/individual practical activity	Group/individual practical activity, Group/individual Laboratory
Session: Independent group work	Session: Independent group work
Resource-based group	Resource-based group
TEL Resource-based group	NOW, submit file
Group practical activity	Group practical activity, group lab

Student group discussion	Peer discussion
TEL peer assessed	Forum
Session: Independent individual work	Session: Independent study
Resource-based individual	Resource-based individual
TEL Resource-based individual	NOW, submit file
Individual practical activity	Individual practical activity, Individual lab
Session: Tutor-individual work	Session: Supervision
One to one coaching	Supervision, One-to-one Tutorial

<u>ATs</u>

ATs LDSE	**ATs SST**
Summative assessment	Summative assessment
Teacher marked summative assessment	Exam, project report, portfolio, presentation, design/create, individual/group practical, practical report, assignment
TEL summative assessment	MCQ, Computer-based assessment
Essay	Essay report, essay exam , open book
Exam	Exam, short answer exam, in class test
Dissertation	Dissertation, project,
Project	Individual project, group project, individual practical, group practical

Appendix [E]: An Example of Design Pattern Data

Design pattern data

Module subject area: BIOL
Module level: 2
Module credit point: 20
Score of S1: 4.7
Score of S2: 4.8
Score of S3: 4.4
Score of S4: 4.8
V1:
V2:
V3:
OverallAlignment:
Lo1- Explain
Lobj1- Explain
TLA1- Lecture
TLA2- Group discussion
Lobj2- Differentiate
TLA1- Resource based group activity
TLA2- Small group discussion
AT1- Individual assignment
AT2- Examination
Lo2- Demonstrate Knowledge
Lobj1- Illustrate
TLA1- Lecture
TLA2- Group discussion
Lobj2- Describe
TLA1- Lecture
TLA2- Small group discussion
AT1- Individual assignment
AT2- Examination
Lo3- Reflect
Lobj1- Discuss
TLA1- Resource based group activity
TLA2- Group discussion
Lobj2- Explain
TLA1- Lecture
TLA2- Class discussion
AT1- Individual assignment

AT2- Examination
Lo4- Analyse
Lobj1- Analyse
TLA1- Lecture
TLA2- Individual practical activity
Lobj2- Justify
TLA1- Group presentation
TLA2- Group activity
AT1- Group assignment
AT2- Examination
Lo5- Evaluate
Lobj1- Apply
TLA1- Seminar
TLA2- Resource based group activity
Lobj2- Analyse
TLA1- Seminar
TLA2- Resource based group activity
AT1- Group assignment
AT2- Examination

Appendix [F]: Verbs, TLAs, and ATs Grouping List

Verbs

Level	Cognitive Ability Stimulated	Action Elicited	Verbs
6	Evaluation	Ability to make a judgment of the worth of something	Argue, Criticize, Evaluate, Justify, Reflect
5	Synthesis	Ability to combine separate part into a new whole or propose alternative solutions	Combine, Produce, Compute, Design, Formulate, Generate, Organize, Summarize, Construct
4	Analysis	Ability to break down objects or ideas into simpler parts and find evidence to support generalizations.	Analyze, Breakdown, Compare and Contrast, Contrast, Differentiate, Distinguish, Predict
3	Application	Ability to apply knowledge to actual situations.	Apply, Assemble, Calculate, Choose, Demonstrate, Estimate, Illustrate, Investigate, Modify, Operate, Prepare, Solve, Implement, Use, Write, Select
2	Comprehension	Ability to understand facts and rephrase knowledge.	Clarify, Classify, Describe, Discuss, Explain, Identify
1	Knowledge	Ability to remember previously learned information	Label, List, Name, Recall, Recognize, Specify, State, Define

TLAs

Group	TLA	Bloom Level
6	Resource Based Group Activity Group Practical Activity TEL Resource Based Group Activity Group Project Group Laboratory Lab	6
4	Resource Based Individual Activity Individual Practical Activity Questioning Example Illustration Laboratory Notebook Individual Project Individual Laboratory Lab Online Exercises	4
3	Student Presentation Class Discussion Small Group Discussion Online Small Group Discussion Group Tutorials Seminars	3
2	Lecture Online Teacher Presentation Supervision Meeting Tutorials	2

ATs

Level	AT
6	Essay Exam Open Book Exam Exam Group Practical Case Studies Portfolio Group Project
5	Individual Assignment Laboratory Notebook Project Report Group Assignment
4	MCQ Individual Practical Individual Project Laboratory Based Assessment Short Answer Exam Practical Report
3	Seminars Presentations Peer Group Presentation Posters

Appendix [G]: t-Test between the Test Set (before) and the Test Set (after)

t-Test: Paired Two Sample for Means		
	V1(Before)	***V1(After)***
Mean	9.606863	6.317647
Variance	27.46975	8.000478
Observations	102	102
Hypothesized Mean Difference	0	0
t Stat	8.770517	
P(T<=t) two-tail	0.000	
t Critical two-tail	1.983731	
	V2(Before)	***V2(After)***
Mean	7.814706	9.730392
Variance	5.758098	3.988352
Observations	102	102
Hypothesized Mean Difference	0	0
t Stat	5.824646	
P(T<=t) two-tail	0.000	
t Critical two-tail	1.983731	
	V3(Before)	***V3(After)***
Mean	10.1549	9.759804
Variance	9.89458	8.911339
Observations	102	102
Hypothesized Mean Difference	0	0
t Stat	2.777646	
P(T<=t) two-tail	0.000	