



Virtual Classroom Simulation:

Design and Trial in a Preservice Teacher Education Program

A thesis submitted by

Simon Skrødal

for the degree of Doctor of Philosophy

School of Education, Faculty of the Professions

The University of Adelaide

June 2010

Supervisory Panel:

SIVAKUMAR ALAGUMALAI, MICHAEL. J. LAWSON,

PAUL CALDER AND ANDREW WENDELBORN

Chapter 1.

Introduction

My personal view of simulation is that it is the most promising ... single innovation ... that we have today. Much, indeed most, of its potential remains to be activated; we have only begun to invent appropriate means for its usage.

(CUNNINGHAM IN CRUICKSHANK, 1969, P. 25)

The research documented in this thesis involves the motivation, conceptualisation, theory, development, trials and evaluation of a computer simulation software; the Virtual Classroom Simulation (VCS). It is the culmination of a study that was first initiated in 2002 by the supervisors in education and the researcher. Expert technology transfer, particularly from areas in the social sciences, conceptual modelling, computer science and mixed method data analysis, has been an instrumental ingredient underpinning the research and development of the VCS. Understanding the trans-disciplinary nature of this work is therefore essential in order to fully appreciate the processes involved in the realisation of this educational simulation. The Venn diagram in Figure 1.1 captures the main areas of the research study.

The thesis is sequenced to highlight the utilisation of computer science design and programming concepts to facilitate and enable educational processes; practicum knowledge pertinent for preservice education programs.

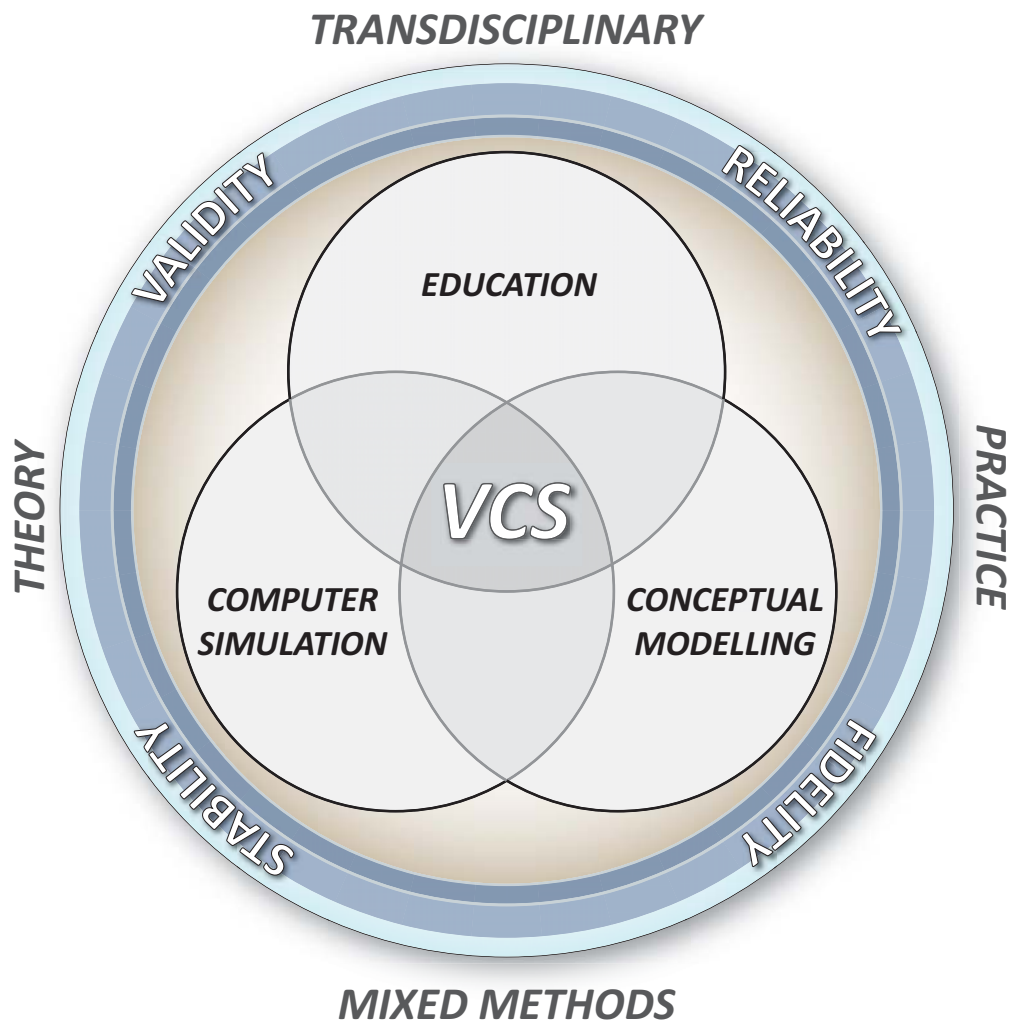


Figure 1.1.: Trans-Disciplinary Composition of the VCS

1.1. Background

A common appreciation for the nexus between pedagogy and Information and Communication Technology (ICT), and particularly how it can aid in teaching and learning, has instigated many interesting discussions between the researcher and supervisors. Some of these have gone on to form research projects and solutions that add value to education. In 2002, a conversation about how simulation software may bridge theory and practice in teacher education led to the conceptualisation of the Virtual Classroom Simulation. Support from a

supervisory panel consisting of two academics in education (Sivakumar Alagumalai and Michael J. Lawson) and one in computer science (Paul Calder) provided the trans-disciplinary expert input essential for advancing the project. A one year study, which commenced in 2003, earned the researcher an Honours degree in Computer Science at the School of Informatics and Engineering, Flinders University (FU), Adelaide, Australia (Skrødal, 2003). The study demonstrated good progress in regard to the theoretical and practical frameworks of the simulation. It acknowledged that the VCS represented the first real-time simulation software of its kind for adding value to preservice teacher education. The research and development documented in this thesis commenced in 2006 and is an extension of the Honours project. Work already established prior to this will, where applicable, be acknowledged throughout this document.

1.1.1. Commencement of the PhD Candidature

Following the Honours candidature the researcher worked as a contractor in the IT industry where he became involved with the innovation and development of a wide range of online and desktop technologies . This experience re-enforced the importance of ‘cross-fertilisation’ between disciplines, as it became apparent that many of these technologies had great potential for educational research and development. It also highlighted how the VCS project could benefit from the convergence of computer science and education in a trans-disciplinary way (see Figure 1.2).

In 2005, the researcher and a small network of content experts in education and science, all with a strong appreciation of ICT in teaching and learning, formally congregated EduPEX (**E**ducation and **P**edagogy **E**xpert Solutions). The organisation pragmatically applies trans-disciplinary skills and knowledge to develop and enhance ICT solutions (services, products and research) for teaching and learning. These activities, and the positive outcomes from the Honours candidature, motivated the expansion of research and development on the VCS. Earlier work had extended the scope and refined the aims and objectives of the simulation with a number of directions for future development. Moreover, the conceptual understanding pertaining to the involved disciplines did also mature in

NOTE:
This figure is included on page 4 of the print copy of
the thesis held in the University of Adelaide Library.

Figure 1.2.: Trans-Disciplinarity (Koizumi, 1999 in OECD, 2002, p. 85)

the period that followed. Considering the evident potential of the VCS, support was sought to advance the research. Firm backing from the original Honours supervisory panel suggested a PhD candidature as an ideal vehicle to continue to research, develop and evaluate the simulation. Whereas the School of Informatics and Engineering at Flinders University hosted the researcher's Honours candidature, the PhD research proposed to operationalise the VCS in a population of education students. The candidature was therefore hosted by the School of Education, University of Adelaide (UoA), now with Andrew L Wendelborn (School of Computer Science, UoA) on board as a fourth supervisor on the panel.

The research was awarded the Australian Postgraduate Award (APA), a scholarship funded by the Australian Government for "students of exceptional research promise who undertake their degree by research at an eligible Australian higher education provider" (The Australian Government Department of Innovation, Industry, Science and Research, 2010).

1.2. Description of the Virtual Classroom Simulation

The VCS is an interactive computer program that aims to assist educators and education students in preservice education programs. It is proposed that, when employed as an integral part of a tertiary education course, the software has the potential to improve education students' practical and theoretical understanding of student-teacher interaction in the classroom. As it may improve their teaching and strategic skills in the classroom, it is a useful resource for practicum preparation. The simulation also addresses a number of challenges in the provision of teacher education. The reasons why it is believed that a simulation like the VCS may be suitable for achieving the above will be substantiated in this dissertation.

1.2.1. How it Works

The software dynamically simulates a range of interactions and behaviours that would typically be found in a real-world classroom context. The graphical user interface (GUI, see Figure 1.3), allows the teacher (i.e. the user) to manipulate a number of program variables (input) and observe consequential changes in the simulated students (output).

Adding to the realistic nature of the VCS, the simulation runs in real-time; i.e. 1 wall clock second maps to 1 simulation second. Carrying out actions such as asking a question or giving feedback carries a 'time-penalty' that is observed in real time by the user, emphasising the importance and dilemmas of efficient time management.

Behavioural variables (or attributes), both visible and latent, of the virtual students include mood, distractibility, self-efficacy, knowledge and meta-cognition. Alterations in these variables are triggered by factors such as the teacher asking a distracted student to pay attention, varying the nature of questions posed to the class (difficulty, novelty, time to ask), variation in the duration before the teacher asks a student, which student is asked (shy students may not raise their hand even though they know the answer), whether or not a student answered the question correctly and if feedback is given. Depending on the actions carried out by the teacher, behavioural consequences may occur in one or more students. Changes in the simulated students are therefore by and large dictated by the 'what', 'how' and

Chapter 1. Introduction

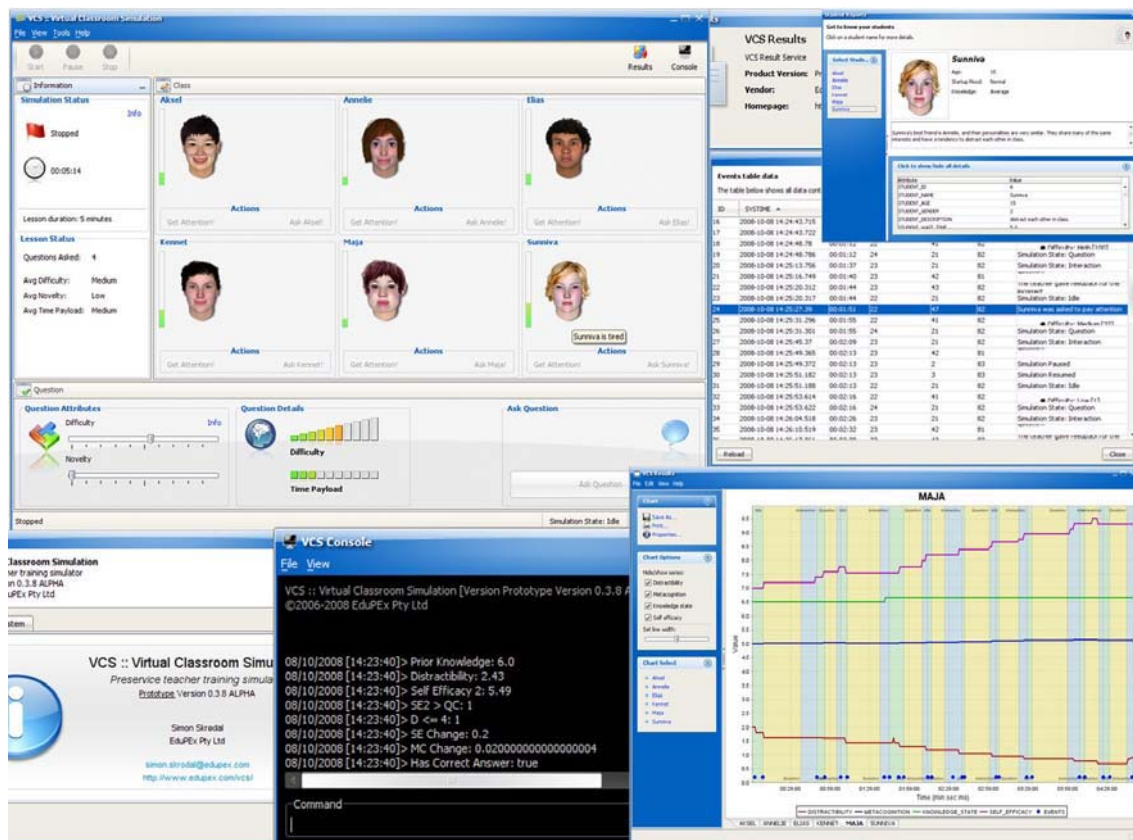


Figure 1.3.: Screen Captures of VCS Windows

‘when’ of teacher interaction. Continuously updated visual indicators, such as facial expressions (mood) and distractibility, are present in order to mirror traits that are also observable in the real-world classroom.

Events are stored to a database as they occur, whilst variable student attribute values are recorded every second throughout the course of the simulation. At the end of a lesson, or episode, captured data are presented back to the users as interactive graphs, allowing them to explore, analyse, reflect and discuss what and how their teaching affected individual students (for example; how asking a very difficult question affected a low-achieving student’s self-efficacy). Reflection, discussion and integration with assessment are important activities that add value to the learning process with the VCS.

The probabilistic and underlying Bayesian nature of the complex algorithms

1.3. Statement of Research Problems

developed for the VCS permits a dynamic, yet well-substantiated, display of behaviours (it is highly unlikely that two identical runs would ever produce the exact same output). The unique simulation architecture was built from first principle and could with little effort be re-used in other social simulation contexts. The following are important characteristics of the VCS and this study:

- While efforts have been made to make the VCS user-friendly (usable), its purpose (usefulness) is not self-explanatory; e.g. it is not obvious to a new user how and why one would use the simulation.
- The VCS is not intended to be used in isolation. Rather, it should be delivered as part of a guided exercise where training and follow-up are integrated and made relevant to the overall objectives of the teacher education course outline.
- This research is, per design, regarded as a pilot study. The VCS software and related material (e.g. surveys and course integration) went through one cycle of comprehensive user trials. It was not proposed that the study would result in a ‘ready-for-market’ system. Rather, it was anticipated that the project would establish and provide insights into how the developed system could be further refined and expanded in future work. While a number of areas for future development have been identified, the VCS in its current state does, however, deliver a range of features and functionalities that add value to the learning process. The final product and user feedback have superseded initial expectations of what would be achieved within the scope of this study.

1.3. Statement of Research Problems

This research identifies a number of challenges in teacher education (especially practicum training). A key objective of the study was to learn how and to what extent a computer simulation could be useful in addressing these challenges. In this instance, the journey from conceptualisation to realisation involved the understanding, development, integration and operationalisation of several constituents. Considering the innovative nature of the proposed study, there were no pre-existing exemplars or guidelines to follow. Hence, instituting the

manageability and feasibility (and thereby the scope) of the research study became a vital exercise at the start of the research. This led to the conceptualisation of three main research problems (Sections 1.3.1 to 1.3.3). These problems, which combined reflect the trans-disciplinary nature of the study, provide the overarching objectives for the research.

The congregation of methods, procedures and techniques pertaining to the discipline of social simulation software development makes up this research study's methodology. A large portion of the research is also concerned with the evaluation of the technical and educational aspects of the VCS with regard to its usability, performance, usefulness and impact. This is embedded in the overall research design for VCS assessment (see Chapter 6).

1.3.1. Research Problem 1

The Honours work had already identified a number of classroom entities and their attributes. While this knowledge would most likely be useful, the objectives and conceptual understandings from prior efforts were not entirely aligned with the current study. Research Problem 1 (RP1) was therefore formulated to address the need for rediscovery, reassessment and re-establishment of constituents required for the study:

Problem 1. What are the most relevant entities in a classroom environment and what are their most significant attributes?

- How can these entities be reconstructed and combined into an efficient interaction model?

Identification of the 'most relevant' entities and attributes in a classroom environment are circumstantial to the objectives of this research. Although prior knowledge was useful, care was made to avoid it obfuscating the advancement of new knowledge. 'Entities' refers to significant classroom components relevant to the student-teacher interaction context. 'Attributes' refers to the entities' properties and/or behavioural aspects relevant to the objectives of this study. The

second part of the problem involves expert insight into relevant educational theory and how to translate specific aspects of a social system into a conceptual model and a computer simulation model. This is largely covered in Chapter 3, which looks at forming an abstract view of the problem and how it can be translated into a simulation context.

The research design of this study included a short questionnaire that aimed to elicit the most relevant factors pertaining to student-teacher interaction, as perceived by a population of preservice teachers (Section 6.3.1, Chapter 6). Questionnaire responses were compared to those from a more detailed survey that followed after the population had a chance to get familiar with the VCS (user trials). The data provide an insight into what the population regarded as important factors of classroom interaction before and after using the simulation. In addressing this problem it is important to realise its significant association with Research Problems 2 and 3.

1.3.2. Research Problem 2

A better understanding of the complexities presented in RP1 is essential before embarking on the following problem. Design decisions must also consider the context in which the software will be used so as to better provide a solution that suits the target user group.

Problem 2. Identify appropriate technologies (hardware and software) that combined allow for efficient and dynamic implementation, management and compatibility of the VCS with regard to:

- input/output (I/O)
- graphical user interface (GUI)
- stability
- speed
- usability

- extensibility
- reliability
- functionality.

Moving into the realm of computer science, the scope of this problem is very broad. It consists of multiple interdependent components that collectively must cater for the other research problems. It demands a good understanding of several aspects in software development, relevant theory in education, target user group, measurement and evaluation.

Assessment of the soft and hard technologies is two-fold. First, the researcher must make informed decisions to determine what technologies are best suitable for the system. It implies the research and assessment of a range of technologies against the criteria stipulated in the research problem. Moreover, this had to be achieved within the restrictions imposed by the research budget (see Section 1.7). Chapters 4 and 5 look at how this was accomplished. Second, the final product needs to be assessed by its intended users (education students) with regard to its usability (technological aspects) and usefulness (educational aspects). To this end, the research design employed two surveys; the VCS HCI (Human-Computer Interaction) Survey (Section 6.3.5, Chapter 6) and the VCS Education Survey (Section 6.3.4, Chapter 6). The former, as the name suggests, seeks to investigate the HCI properties of the simulation software. It is composed by the adaptation of two questionnaires; the Computer Familiarity Instrument from the PISA 2000 Study (OECD, 2000; Adams and Wu, 2002, pp. 242-243) and the Questionnaire for User Interaction Satisfaction (QUIS, Chin et al., 1988). The VCS HCI Survey provides an insight into the population's experience and attitudes towards ICTs in general as well as their satisfaction with the usability of the VCS. While the VCS Education Survey provides some further details to the usability of the VCS, its main focus is concerned with the simulation's usefulness for teaching and learning.

1.3.3. Research Problem 3

The scaffolding needed to understand the social processes of the real system (what Skrødal et al. (2006) call the “observe — deconstruct — identify — construct — develop — simulate hierarchy”) must give careful consideration to all three research problems. The third and final problem is concerned with evaluating the materialisation of problems 1 and 2. It lies at the heart of the objectives of the study and was therefore fundamental throughout the design and development stages of the system as well as the research instrumentation.

Problem 3. To what extent is the VCS useful in reshaping preservice teachers’ mental model about teaching and learning, as well as advancing their pedagogical content knowledge?

- How can this be accurately measured?
- How can the impact of the VCS be further improved?

A proficient understanding of the epistemology in education, as well as a good command of relevant principles in computer science, are essential in order to establish what aspects to include in the simulation and how these are to be implemented. However, while this problem cannot be dissertated without reference to the previous research problems, it also introduces a number of new constructs in need of consideration. These constructs, which make up the research design, are fundamental for devising the methods that enable an adept evaluation of the VCS and how it may assist in the transfer of knowledge. This research study follows a mixed-method approach that employs a carefully chosen design to analyse and substantiate the reliability and validity of the simulation and its effect on learning.

Chapter 6 provides more details about the research design and includes a detailed account of user trials, instruments, data collection and data analyses.

1.4. Participants (Population)

The VCS was designed to add value to teacher education and user participation and feedback were essential for addressing some aspects of the research problems. The School of Education at the University of Adelaide offers two pathways to become a qualified school teacher; an undergraduate degree in Bachelor of Teaching (BTeach/B.Arts, B.Science, B.Economics or B.Music Education) and a postgraduate Graduate Diploma in Education (GradDipEd). The former is offered as a 4-year double degree with Arts, Economics, Mathematical and Computer Sciences or Science, whilst the latter is a one-year programme available for students who have already completed an undergraduate degree. This research study aimed to cover the entire population, i.e. all students enrolled in the 4th year BTeach and the GradDipEd programme in semester 1, 2008 (see Table 1.1). The population was considered most suitable since the VCS aims to assist in the education of student teachers.

Table 1.1.: Population and Gender Distribution

	BTeach	GradDipEd	Total
Female	38	79	117
Male	23	55	78
Total	61	134	195

Firm support from the School of Education allowed for the VCS to be incorporated as an integral part of “Student-Teacher Interaction in the Classroom 1” (STIC); a compulsory course for all 4th year students (course outline available in Appendix I). This particular course is highly appropriate for VCS integration, as it covers many of the theoretical aspects pertaining to the VCS. Support from the course coordinator (Dr. Nina Maadad) and assistant lecturer (Mr. Alan Larkin) allowed for the researcher to administer a pre-trial questionnaire, an introductory lecture about the VCS and a 50-minute tutorial in which a hands-on VCS trial and two surveys could be administered. Moreover, the course coordinator and the researcher also devised a graded assignment that used concepts from the VCS and, importantly, education and praxis knowledge of student-teacher interaction in the classroom.

1.5. Theoretical Framework

A theoretical framework, which employed a mixed set of methods, was constructed in order to address the research problems. It was established on the basis of the instruments employed in the research design of this study. Of particular interest to the study was an investigation into the causal effects between the population’s skills and attitudes towards computers, their overall reactions to the usability of the VCS and their perception of the simulation’s usefulness in education. A simplified version of the proposed model for analysis, as shown in Figure 1.4, includes a number of factors involved in the research instruments. These will be addressed in more detail in later chapters.

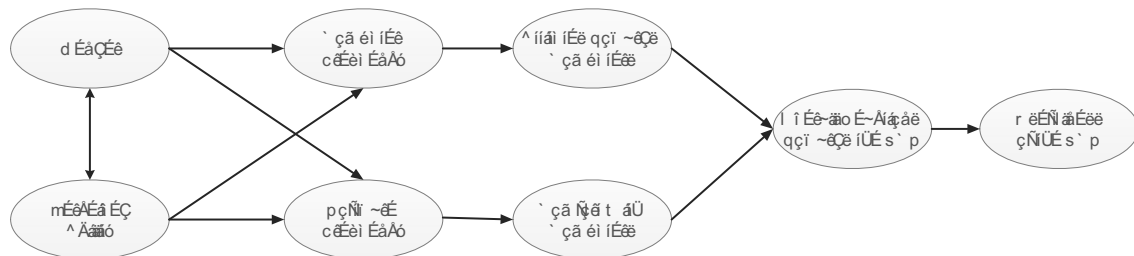


Figure 1.4.: Abstract View of the Theoretical Model

Specifically, Chapter 6 describes the research design of this study. Chapters 7 to 9 detail the instruments employed to collect data and establish their reliability and validity. Findings produced by the triangulation mixed methods design (Creswell, 2008) were instrumental in addressing the research problems and the final evaluation of the VCS’ capacity to adjust and advance education students’ mental models in the teaching and learning context. These findings are reported in Chapter 10, which will discuss the final outcome of the analysis.

1.6. Timeline for Research Study

Figure 1.5 gives a broad overview of the activities undertaken in this study.

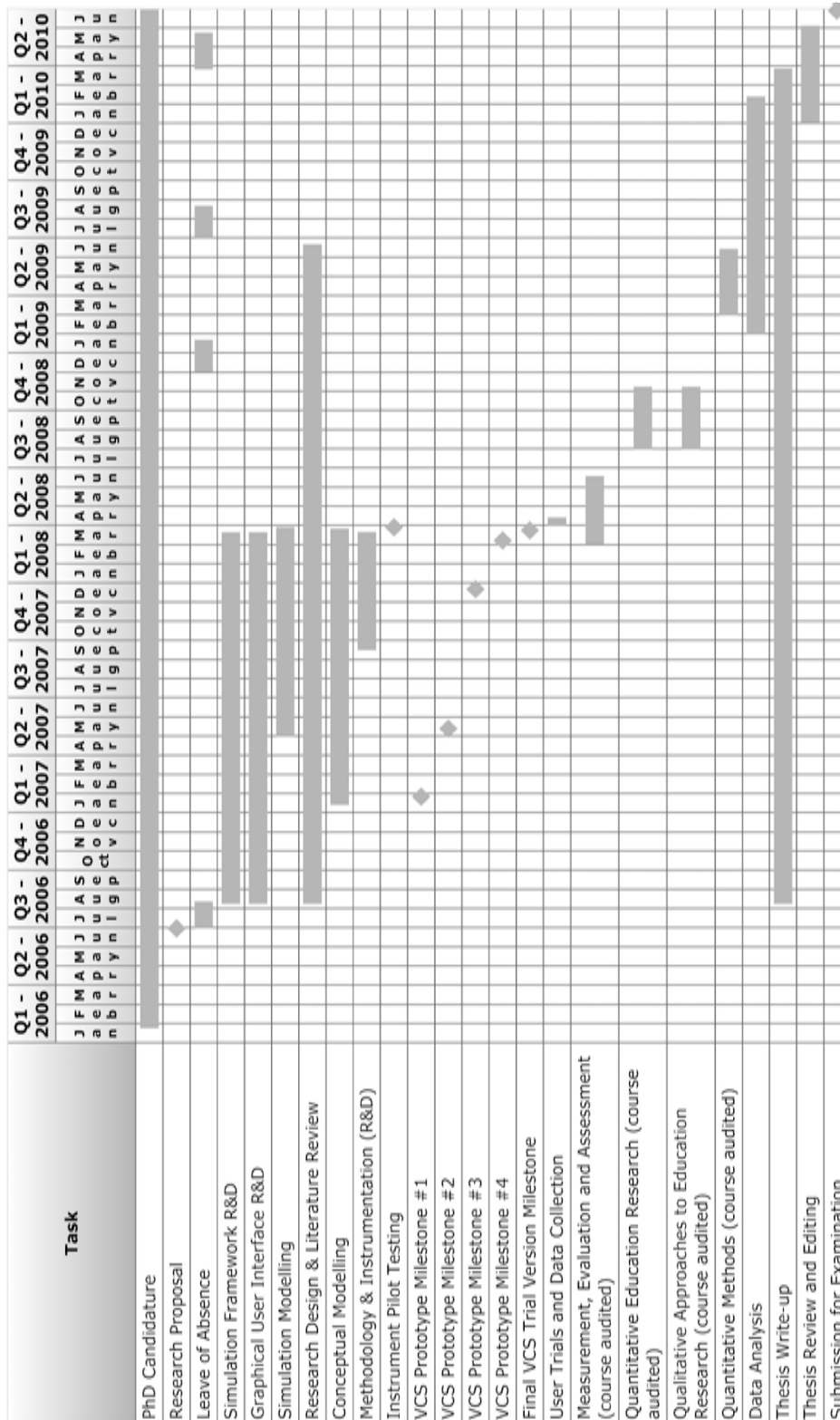


Figure 1.5.: Research & Development Timeline

1.7. Research Budget

This research study operated with a limited budget. Thus, no financial resources were used towards the acquisition of new software, components, books, manuals or other research assistance to support development, data collection or reporting. Instead, much effort was invested in sourcing the most suitable free alternatives. Some resources were already made available by the university, such as library access and licensed access to journal articles online and various software (e.g. for data analysis). For a complete list of software and components used in this study, please see Appendix A.

1.8. Intellectual Property

To the best of the researcher's and the supervisory panel's knowledge, no research or software similar in nature to what is proposed by this research exist. With the amount of resources invested in the project over a number of years, consultations with legal experts highlighted the importance of Intellectual Property (IP) protection. Appropriate management of IP was therefore established prior to commencing the PhD research study. Although this has, to some extent, limited the number and nature of VCS-related publications, the research has received interest from a number of organisations world-wide (see Appendices U and V for examples).

1.9. Summary

The conceptualisation of the Virtual Classroom Simulation, a computer software to assist education students, dates back to 2002. Research and development of the software was advanced with the researcher's Honours candidature in 2003. Experiences and findings from these efforts suggested that the project had, if further developed, strong potential to add value to preservice teacher education. The current PhD research commenced in 2006 with support from the Australian Government. A Virtual Classroom Simulation was developed from first principle and trialled on a population of preservice teachers. The simulation's capacity to

Chapter 1. Introduction

reshape education students' mental models about teaching and learning was evaluated using a triangulation mixed methods design and advanced techniques for establishing instrument reliability and validity were employed prior to the final analyses of data.

The next chapter will provide further insights into the background and motivation of the study documented in this thesis. Chapter 3, which addresses RP1 of this study, discusses the abstraction and design of a classroom interaction context. Findings from this chapter will inform the development of the simulation framework (Chapter 4) and the graphical user interface (Chapter 5).

Chapter 2.

Background

Once beliefs are formed, individuals have a tendency to build causal explanations surrounding the aspects of those beliefs, whether these explanations are accurate or mere invention.

PAJARES (1992, P. 317)

Research Problem 3, which is a cornerstone of this research, cannot be addressed from the viewpoint of a single discipline. Facets from a number of disciplines involved in the study, e.g. education, psychology, social sciences, conceptual modelling, computer science and underlying mixed methods research design, must be taken into account.

Problem 3. To what extent is the VCS useful in reshaping preservice teachers' mental model about teaching and learning, as well as advancing their pedagogical content knowledge?

- How can this be accurately measured?
- How can the impact of the VCS be further improved?

However, educational theories, practices and challenges underlie the design of the Virtual Classroom Simulation (VCS). As this chapter will demonstrate, research and development of the simulation was motivated by a number of challenges identified in preservice teacher education. The proposed solution to some of these problems is

not necessarily the invention or discovery of new educational theory, but rather the use of existing knowledge in new and innovative ways. While it is the integration of multiple disciplines that has made the VCS an innovative tool for teaching and learning, the end product is nonetheless conceived out of and aimed at education.

2.1. Challenges in Education

This study proposes that a number challenges in preservice teacher education may be mitigated with the aid of computer simulation technology. The following subsections provide an insight into the nature of these challenges, as well as why and how they may be alleviated by the VCS.

2.1.1. Mental Models, Preconceptions & Schema

...everyone forms theories (mental models) to explain what they have observed. [...] In the absence of external information, people are free to let their imaginations run free as long as the mental models they develop account for the facts as they perceive them. Norman (1988, p. 39)

A number of research studies in education highlight what attributes effective teachers bring to the classroom. Anderson (1989) argues that, in order to successfully fulfill their many roles, teachers often resort to personal theories and conceptions of education when establishing a good learning environment. Both Anderson (1989) and Barnes (1989) suggest that there is a large knowledge base available to support beginning teachers, which includes knowledge about learners, learning and classroom instruction. However, as Barnes (1989) indicates, teacher education programmes too often tend to focus on what education students should know in terms of their content knowledge, rather than what they think they already know. Shulman (2000, p.64) lists a number of categories belonging to the knowledge base, which include:

- content knowledge
- pedagogical knowledge

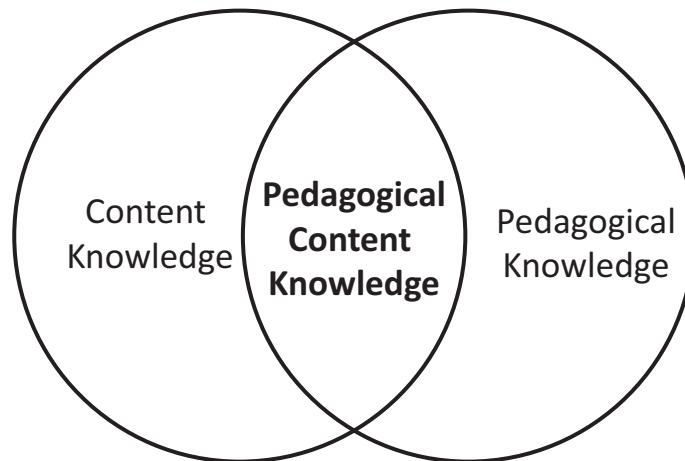


Figure 2.1.: Intersect of Content and Pedagogy (based on Shulman, 2000, p. 64)

- curriculum knowledge
- pedagogical content knowledge
- knowledge of learners and their characteristics
- knowledge of educational contexts
- knowledge of educational ends, purposes and values and their philosophical and historical grounds

One area of special interest to Shulman (2000) is the intersect between content and pedagogy; a category he labels *pedagogical content knowledge* (PCK). This intersect, as illustrated in Figure 2.1, is described by Shulman (2000, p. 64) as “that special amalgam of content and pedagogy that is uniquely the province of teachers, their own special form of professional understanding”.

According to Shulman (2000), PCK allows for a better understanding of how to cater for the diverse interests and abilities of learners. In a pedagogical sense, having a knowledge of different strategies (not to be confused with Shulman’s notion of curricular/strategic knowledge) of teaching can be essential in reorganising the understanding of learners, whose preconceptions are frequently misconceptions (Shulman, 1994).

Chapter 2. Background

The transfer of theoretical aspects of teaching into practice may be hindered by the lack of integration between what Corporaal calls “the teacher educator’s theory” and “the student teacher’s theory” (Corporaal, 1988, cited in Korthagen and Kessels, 1999). Research has unveiled that *most* preservice teachers keep an idealistic view of their own competencies, leading them to believe that they are unlikely to face the same problems as other teachers (Pajares, 1992). A number of student teachers also believe that they will make better teachers than their peers (Pajares, 1992), and a major concern in this regard is that such beliefs are not likely to change much without major intervention (Cooney et al., 1998). In a report prepared for the Australian Minister for Education, Fry and Flashman (2005, p. 7) argue that preservice teacher training “needs to be able to identify arising disappointment, possible mismatches and distress between actual and ideal self-image”.

Evidence shows that even experienced teachers tend to have a limited understanding of their students’ learning preferences, and that their own preferred way of learning is often projected onto their students (Korthagen and Kessels, 1999). In their education, preservice teachers should therefore be forced to challenge their own ‘mental models’ (Johnson-Laird, 1983), or ‘mental concepts’ (Armstrong, 2004), about teaching and learning as their schema are often based on their life long experience and world-view as students (Mellado, 1998). Failing to deal with these preconceptions appropriately, which is what Barnes (1989) suggests is happening at large, may provoke undesired consequences for new teachers entering the profession. According to Fry and Flashman (2005), for example, there is much literature to suggest that beginning teachers in Australia feel unprepared and unsupported when entering the workforce. Consequently, a staggering 20-25% burnout rate is observed in beginning teachers’ first three years in service (Fry and Flashman, 2005).

2.1.2. Practice Teaching

Many teacher programmes have been criticised for not linking theory with practice (Barone et al., 1996 in Korthagen and Kessels, 1999). In order to provide hands-on experience for education students, teaching programmes commonly include a school placement component (also known as practice teaching or practicum). This

2.1. Challenges in Education

component provides a bridge between theory and vicarious practice that most students rate as being more valuable than face-to-face lecturer -or tutorial-driven courses (Skrødal and Alagumalai, 2009; Bencze et al., 2001). Practice teaching is an important part of the teacher education curriculum, as it gives the beginning teacher a relevant perspective of the studied theory. Not only does this allow them to draw on experience from expert mentor teachers, as well as gaining knowledge about the workings of the education institution; it also gives them a unique opportunity to teach students under qualified guidance and supervision. The prospect of advancing pedagogical content knowledge through first-hand teaching experience is significant, and although the exact nature of the teaching practicum (timing, duration, location and integration) is not always subject to unanimous agreement, its added value is unquestioning (Taffe and Knipe, 2005). In fact, some research has shown that the practice teaching component is the only part of teacher training that has a considerable influence on education students' existing conceptions about teaching Mellado (1998). Similarly, Stones (1987) argues that practice teaching is commonly considered to be the single most important component of teacher education.

One would, based on the above, presume that a university programme component deemed as valuable as practice teaching would be utilised to its full potential. Partially, this may be so, considering that it is a requirement for Australian universities to provide practicum placements. Before students can register as teachers they must complete 45 to 80 days of supervised teaching (DEEWR, 2008). On the other hand, schools are not obligated to offer such places to the universities. Unfortunately, the mismatch between number of education students and available practicum placements has caused what Leung (2006) labelled as 'a crisis' in teaching placements in Australia. This was echoed in a comprehensive report issued by the Australian Standing Committee on Education and Vocational Training, where the shortage of practicum placements for secondary schools was labelled as 'particularly serious' (Hartsuyker, 2007, p. 70). In a recent South Australian newspaper article, the Deans of Education from all three universities in South Australia expressed their concerns about practicum placements (Novak, 2009, p. 29):

Chapter 2. Background

We are having tremendous difficulty in trying to find placements.

A/Prof. Sivakumar Alagumalai (Head of School of Education, University of Adelaide)

...we are experiencing some serious shortfalls in the number of placements in schools...

Prof. Bob Conway (Dean of Education, Flinders University)

Finding student places has been a challenge for a long time.

A/Prof. Michele Simons (Head of Education, University of South Australia)

In addition to these difficulties, issues with finding suitable mentor teachers complicate things further; i.e. to match students to schools that offer specialist subjects. Professor Conway explains:

We don't want to just put students anywhere, we want to put them in classrooms where it will be a worthwhile learning experience. (Novak, 2009, p. 29)

A similar story was reported in the media again in 2010, with the Deans of Education suggesting the situation is getting more difficult each year. In 2010, the School of Education, University of Adelaide, had to find over 100 extra placements compared to what was the case the previous year (Hood, 2010).

In 2007, the Australian Government also reported a concern about the weak link between theoretical components of teacher training and the practicum, as well as a decreasing level and quality of financial and intellectual/professional support:

The problems with practicum have been outlined in nearly every report addressing teacher education in the last decade. The fact that these problems have still drawn so much attention in this inquiry indicates the need for major reform in this area, involving all players and all aspects of the system. (Hartsuyker, 2007, p. 73)

Le Cornu (2005) provides further insight into the myriad of related issues (e.g. shortage of placements, students' circumstances, administration, politics, economy,

accountability and standards) faced by universities in South Australia. Fry and Flashman (2005) point to research which informs that student teachers do not receive formal training on classroom management. Maclean and White (2005, p. 302) go so far as to say that “there is an urgent need for a better way of using school placements to prepare student teachers to enter the profession”.

2.1.3. Reflective Practice

While there is no shortage of publications to suggest what can be improved in teacher education, some do also offer relief through advice on how to address these issues. Reflection, or *reflective practice*, an approach that picked up a lot of momentum throughout the 1990s, is agreed by many to be of much value in the education of student teachers. Although reflection is a complex process which makes it difficult to teach, research suggests that the method is useful in helping education students to “think like a teacher” (Jay and Johnson, 2002). It is believed that learning becomes more explicit through the process of critical thinking and reassessment of own beliefs and practice. Thus, it has the potential to enhance education students’ problem solving skills and their understanding about the teaching practice (Clarke, 2005). Fry and Flashman (2005, p. 8) stress that theoretical knowledge “derived from philosophical positions should remain as a foundation of teacher training” and advocate the importance of reflective practice in this context.

Exactly what constitutes reflection, however, and how it contributes to the education of teachers, is an area of ambiguity (Loughran, 2002; Maclean and White, 2005). Maclean and White (2005) present two views about the nature of reflection. Firstly, reflection can be perceived as the adjustment of personal theory (e.g. beliefs about teaching and learning) as a consequence of insights into critical events. Secondly, reflection may also be considered a social process where a team negotiates shared tasks and understandings, where the collective output takes the form of revised future goals for action and/or routines. Shulman describes reflection as the processes of learning from experience; processes that may include “reviewing, reconstructing, reenacting and critically analyzing one’s own and the class’s performance, and grounding explanations in evidence” (Shulman, 2000, p. 71). Hoffman-Kipp et al. (2003) insist that reflection should be seen as a practice

embedded in a larger learning process; praxis.

2.1.4. Praxis

The definition of praxis from Hoffman-Kipp et al. (2003, p. 249) as the “dialectical union of reflection and action”, or Chinn and Kramer (1999, cited in Skrødal and Alagumalai, 2009) as a “thoughtful reflection and action that occur in synchrony”, is further substantiated by Breunig (2005, p. 111):

Praxis, therefore, starts with an abstract idea (theory) or an experience, and incorporates reflection upon that idea or experience and then translates it into purposeful action. Praxis is reflective, active, creative, contextual, purposeful, and socially constructed.

Bencze et al. (2001) speak of the chasm, or ‘theory-practice gap’, that exists between academic teacher education and what happens in schools. They suggest that the academic and teacher practicum components alone do not suffice, as student teachers often revert back to their own mental models. While overcoming these barriers is difficult, Bencze et al. (2001) propose that getting education students into the habit of *praxis* may be of value. Specifically, they suggest that this can be achieved with the aid of multimedia-enhanced case methods that challenge student teachers to become ‘reflective practitioners’. Praxis, then, may be used to describe the mesh of theory, mental models, reflection and practice that, combined, equates to a more proficient learning process and improved adjustment of misconceptions.

2.1.5. Student-Teacher Interaction in the Classroom (STIC)

“Student Teacher Interaction in the Classroom” (STIC) is a course offered in the education program at the School of Education, University of Adelaide. This core-component subject is held prior to the education students’ practicum placements. STIC “seeks to expound the understanding of advanced practice in education” and to examine “the various forces and dimensions at work in the classroom, including the interactions between students, the student and the teacher, the task at hand and the learning environment” (Skrødal and Alagumalai, 2009, p. 3).

2.2. Alternative Solution: Simulation

Although educational theory and reflective practice is not lost in the STIC course, the constitution of praxis through their amalgamation with ‘purposeful action’ is missing. Hence, practice teaching is a much needed constituent to complement the course material. However, as discussed in Section 2.1.2, a number of challenges have been linked to the practicum component. Bencze et al. also argued that while the practicum provides the most authentic conditions, “that too is insufficient, given student teachers’ transient position in the teaching/learning environment” (Bencze et al., 2001, p. 196). Further, despite the many complexities in demonstrating praxis for education students, it should ideally be attainable within the context of a course like STIC prior to the practicum placement. For this to be achieved alternative approaches should be considered.

2.2. Alternative Solution: Simulation

While contemplating on the future of practice teaching, Stones (1987) recommended that it would be even more valuable to couple the teaching placement with a greater emphasis on self-reflection and discussion between preservice teachers. Stones (1987) argued that expanding this component would assist in developing a better understanding of the body of pedagogical theory and, consequently, increase their pedagogical skills. In order to foster this, Stones (1987) indicated that the time devoted to practice teaching could in fact be reduced to allow more time for informed and effective reflective activities. Barnes (1989) supports the argument that practical experience in the classroom alone will not suffice in fostering preservice teachers’ much needed expertise, nor in challenging their potential misconceptions about teaching. Thus, one could argue that even with a sufficient number of practice teaching placements, there is still a need for extra components in the teacher education program. One promising approach suggests a program that is “conceptualized and designed to challenge unexamined assumptions about children, learning, subject matter, and teaching, and to promote a new understanding of learning and teaching” (Barnes, 1989, p. 19). Hence, coupling the lack of available teaching placements with Stones’ (1987) and Barnes’ (1989) notions of shortcomings in teacher training, it seems appropriate to propose a new paradigm in this field. Griffin (1989, p. 282)

Chapter 2. Background

suggested that:

The character of classrooms and schools demands a thoughtful and deliberate seeking out of ways of thinking and acting. And the search must consider alternate paradigms, competing propositions, and multiple interpretations of reality.

An alternative perspective in the education of student teachers that should be considered is the use of computer technology and simulation models. For example, computer simulations as modelling and training tools have already been utilised in numerous institutions and organisations for decades; flight simulators for pilots, market simulations for stock brokers and weather forecasting systems for meteorologists, to name a few.

In 1969, when it was announced that American airlines were about to make major investments in computer simulations for pilot training, Cruickshank (1969) made a remarkable connection between simulations and their potential for teacher education. It is an astonishing read, because in this article Cruickshank (1969) sums up many key arguments, motivating factors and opportunities that are true for the research and development of the VCS today; e.g. quality and availability of teacher practicum placement, combination of theory and practice, intellectual and economic benefits for both the education student and the university, safe and failure-free environment for training and experience, how to approximate likely student behaviour, too little research evidence on viability of simulation, where and when to use the simulation and assessment of learning benefits.

...it is possible, for example, to have the trainee encounter in just two weeks the most critical problems he will face in his first year of teaching, in a threat-free, failure-free environment unlike that of student teaching.
(Cruickshank, 1969, p. 24)

Thus, the use of computer simulations to assist in teacher education is not a recent concept. However, the advancement from concept to conceptualisation to realisation has arguably not made as much progress as Cruickshank may have anticipated in the 1960s.

In recent years, research and products with titles akin to that of the VCS have come into existence. In particular, 'Virtual Classroom' has become a generic term

which encompasses online applications for teaching and learning that remove geographic barriers (Yang and Liu, 2007; Bower, 2006; Arbaugh, 2000). Often incorporated into a Learning Management System (LMS), these web-based environments may include features such as video conferencing, multimedia presentations, forums, chat, tests, assessment, content management and so forth. LMS and related technologies can be useful enablers of different modes of interaction and content sharing between people in the teaching and learning context. It can also be argued that some of these virtual environments do, to some degree, simulate a real classroom environment. However, if the term ‘virtual’ is taken to mean something that is not concrete or physical, only the intermediate medium (e.g. the learning environment) is virtual; the people on both ends are still real. Further, the simulation qualities of these virtual environments are strictly limited to the mimicking (imitation) of tools and practices found in the real environment (e.g. virtual whiteboard, forums, etc.); the processes of the system are still the product of user interaction. Helsgaun (2000, p. 3) defines simulation as “the experimentation with a model in order to obtain information about the dynamic behavior of a system”. The objectives for, and thereby the features and functionalities of, a virtual *simulation* of a classroom are thus substantially different to those of a system facilitating for human interaction and content management.

2.2.1. Existing Simulations

Little research has been done on simulations for teacher education, but there are some contributors to the field. Ferry et al. (2006; 2004) challenge the traditional education programmes and claim that they are inadequate in developing a number of key skills needed by education students. In accordance with the VCS study, they too propose that a simulation system could be useful in supporting teacher education. Ferry et al. (2006; 2004) developed a web-based simulation prototype to address these issues. However, the pedagogical focus (teaching of literacy skills), target audience (kindergarten teachers), method of delivery (web-based) and simulation type (deterministic) differ significantly from what is proposed for the current study. Their prototype takes the shape of a ‘walk-through’ of a sequence of screens in

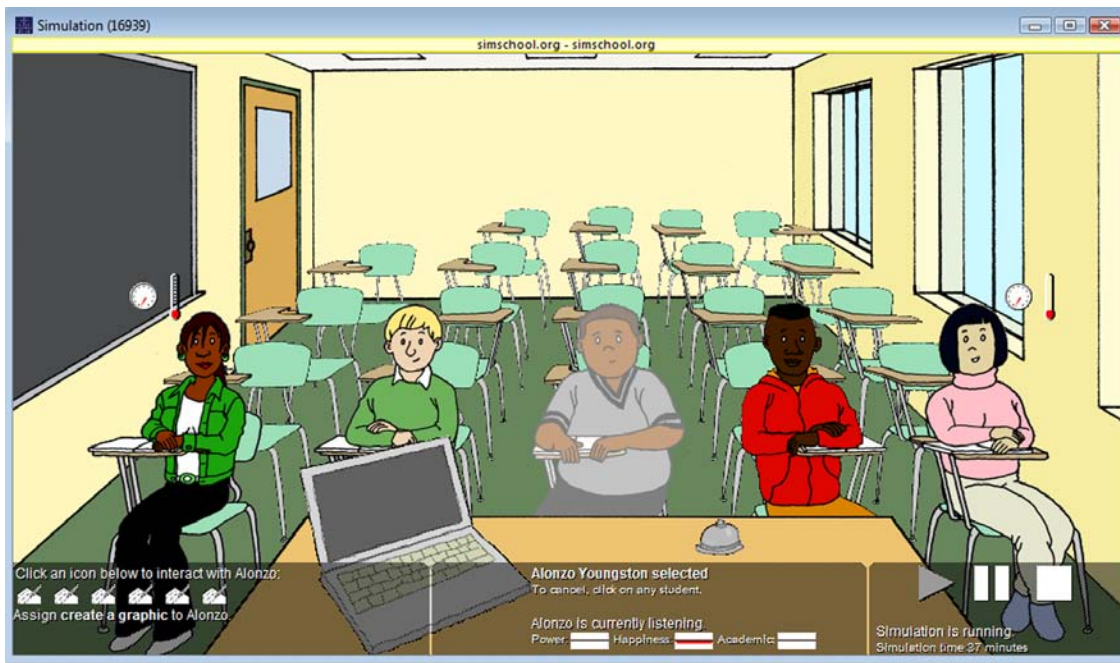


Figure 2.2.: Screen Capture of the simSchool Simulation

which the teacher (the user) is faced with making a number of decisions related to classroom-management and organisation (Ferry et al., 2006). The simulation type may be labelled as ‘deterministic’ because it appears that a given input always leads to the same output.

Another simulation of interest, ‘simSchool’, came into existence a few years after the VCS. simSchool is an application jointly developed by educators, researchers and programmers (Gibson, 2008). Referred to as a ‘virtual practicum’, this simulation aims to help education students better interpret student behaviour and academic skills (Zibit and Halverson, 2006). The trial version, which may be accessed free of charge on the simSchool website (<http://www.simschool.org>), reveals features of the simulation. A large portion of the source code (JavaScript) can readily be accessed with most modern web browsers (using the ‘View Source’ or similarly named command).

The simulation provides a number of interaction opportunities that can be applied to the class as a whole or to individual students. While the VCS tracks student attributes of ‘knowledge state’, ‘mood’, ‘distractibility’, ‘self-efficacy’ and

‘metacognition’, simSchool focuses on ‘academic agreeableness’, ‘emotion’, ‘extroversion’, ‘intellect’ and ‘persistence’. In both simulations, many of these attributes are hidden from the user by default (although they can be accessed in the VCS) and student report cards (or ‘profiles’ in simSchool) containing narrative descriptions of each student are made available to the user instead.

Both the VCS and simSchool make use of interchangeable static images to represent different moods (or attitudes) of the students. Similar to the distractibility meter in the VCS, simSchool also utilises indicators where attributes cannot be visualised directly in the student pictures (see Figure 2.2).

At the end of the simulation, simSchool provides charts (static images) for each student that display attribute changes over time (30 second blocks) and, to some extent, simulation events. The VCS also provides result charts, although these are interactive (zoom, pan, clickable areas for event details, inclusion/exclusion of attributes and appearance of charts) and provide more detail (per-second resolution, interaction states and event details). Both simulations are platform-independent (can run on any operating system). simSchool is web-based and depends on a JavaScript-compliant Internet web browser. The VCS is a desktop application that needs the Java Runtime Environment to operate.

The researcher was unable to locate published work related to how simSchool is supported by research. Moreover, it is not known if the simulation has been involved in any trials or evaluations. Thus, no data are available to inform about the educational usefulness of the simulation. It is nonetheless encouraging to observe that other projects on simulations for teacher education exist. As this thesis will demonstrate, the VCS sets itself apart from other solutions in a number of areas.

2.2.2. Emerging Support for Simulations in Teacher Education

Recently in Australia, the Strategic ICT Advisory Service (SICTAS) was commissioned by the Department of Education, Employment and Workplace Relations (DEEWR) to deliver a report on ICT in preservice teacher education. One area of particular interest is the investigation of preservice teachers’ experience in applying ICT in pedagogy. In the executive summary, Black et al. state:

Chapter 2. Background

The evidence presented in this paper strongly points to fundamental systemic flaws in the pre-service teacher education system in Australia in terms of developing teacher competence in embedding ICTs in pedagogy and practice. (Black et al., 2009, p. 7)

The report lists a number of recommendations to alleviate these ‘systemic flaws’. One that holds relevance to the VCS study is the notion of a ‘suite of virtual world schools’. Black et al. (2009, p. 34) suggest that such a suite “provides an opportunity to transform the practicum through the use of virtual world simulations so that student teachers are able to experience ‘real’ teaching situations where the newly developing pedagogy could be practiced and assessed”. Similar to what has already been emphasised in the VCS study, Black et al. (2009, p. 34) imply that the development of a simulated school requires a trans-disciplinary approach in which teacher academics, teachers and preservice teachers are involved.

Shulman (1991) stressed that the pedagogical ways of knowing encompass subject matter and its representation, as well as the process of learning and teaching. In terms of these pedagogical ways of knowing, the VCS falls within the scope of learning and teaching. It highlights the differences between individual preferences of learning and the impact that varied styles of teaching may have on students. The VCS also demonstrates the potential differences that exist between education students’ theories and what happens in practice. As a hands-on component of teacher education, the simulation allows the education students to contrast their perceptions with what is observed in the virtual system in a continuous feedback process (Mellado, 1998, p. 210).

2.3. Summary

This chapter has identified a number of challenges in teacher education:

- Education students’ mental models (preconceptions) of teaching and learning,
- Education students’ idealistic view of their own competencies,
- Content vs. pedagogical knowledge,

- Practicum placements,
- Utilisation of reflective practices.

An underlying issue in teacher education appears to be how education students' preconceptions often hinder or distort the transfer of theoretical and practical know-hows. Another problem in teacher education is what Korthagen and Kessels (1999) refer to as the poor transfer of theory into practice. The VCS may be thought of as an 'instigator of praxis', as it establishes the nexus of educational theory, hands-on practice and reflection.

This research set out to develop a simulation software with the aim to assist education students challenge their mental models of teaching and learning. From this process, they will also be disposed towards reassessing their own competencies. The educational theories underpinning the VCS are aligned with those taught in tertiary education courses. The simulation therefore provides a practical approach to re-emphasising, learning and understanding the pedagogical implications of these bona fide theories through hands-on experience and reflective practices.

The next three chapters are concerned with the development of the VCS software. Specifically, in Chapter 3 the real-world classroom will be deconstructed, abstracted and reconstructed to form a conceptual model to be implemented in the software. Chapter 4 accounts for the simulation framework and how it was designed to facilitate the conceptual model, interaction algorithms and logic, data storage and user interaction. The latter involves the provision of features that allows a graphical user interface to interact with the framework. The graphical user interface, which is a communication layer between the simulation framework and the user is detailed in Chapter 5.

Chapter 2. Background

Chapter 3.

Conceptual and Simulation Modelling

Since we cannot comprehend completely and absolutely a society (nor any phenomenon), in order to understand how it functions, we have no other choice than to make abstractions of "the real thing"

GERSHENSON (2002)

Gershenson's introductory quote is particularly relevant in regard to conceptual modelling, since this is where the abstractions of what he refers to as 'the real thing' take place.

Figure 3.1 provides an applicable overview of areas and processes involved in this study.

The previous chapter aimed to provide a better understanding of the theory behind the 'problem entity', or the 'real thing', studied in this context by offering an insight into its purpose, problems and possible solutions. This chapter focuses on the processes involved in making relevant and valid abstractions of the problem entity and the use of this information to create a conceptual model. The conceptual model establishes the basis of properties and logic needed to form a computerised model to be implemented in the simulation software. In other words, of the three disciplines illustrated in the Venn diagram in Figure 1.1 (page 2 in Chapter 1), *conceptual modelling* may be considered to be what facilitates the translation and transfer of a real system in *education* into a subset of parameters (a model) to be utilised in a *computer simulation*. Figure 3.1 also highlights the concurrent cross-validation that needs to occur between the problem entity and the conceptual and computerised models in the development stages. Chapter 6

NOTE:
This figure is included on page 34 of the print copy of
the thesis held in the University of Adelaide Library.

Figure 3.1.: Simplified Version of the Modelling Process (adapted from Sargent, 2005, p. 132)

onwards demonstrate the final analysis, validation and evaluation of the VCS, which places much emphasis on feedback from a target user population.

3.1. Terminology

Many, sometimes conflicting, definitions and conceptions exist in the area of modelling and simulation (Becker et al., 2005). Although some may be attributed to inter-disciplinary differences, for example between the natural sciences and the social sciences, domain-specific contrasts do also exist. To avoid confusion of meanings of definitions and intentions, this section provides understandings subscribed to by this research. It will also demonstrate how different constructs fit together to form what is in this research a complete model for computer simulation.

System (a.k.a. Domain, Process, Problem Entity, Phenomena) — A system typically constitutes a set of interconnected components, processes and subsystems situated within a specific context/scenario. Although labelled by Gershenson (2002) as “the real thing” it should not be taken too literally, as a system may be real or proposed (Sargent, 2005). In theory, any system,

real or artificial, can be modelled and simulated, although not always in its entirety if the system is very complex (e.g. social systems). Therefore, a transparent and unambiguous specification of the simulation *objectives* is useful to help reduce system complexity, or rather deduct aspects of the system irrelevant to the set objectives (*abstraction*). This research, for example, aims to model a system of real-world student-teacher interaction in the classroom. The boundaries of what is (and what is not) to be included is clearly defined, and aspects (entities and processes) of the system that fall within these boundaries may also be further reduced.

Entity (a.k.a. Object, Component) — Exactly what constitutes an entity largely comes down to the contextual definitions made by the modeller/developer. Sargent (2005), for example, considers the system itself as a “problem entity” while Chavalarias (2006) refers to an entity as the artificial representation of a human being. Joines and Roberts (1999, p. 133) use the terms “object” and “component” interchangeably as an entity that “holds both the descriptive attributes of the object as well as defines its behavior”. Similarly, Helsgaun (2000) sees the entity as having zero or more attributes to describe its state, whilst the collection of all attributes describes the state of the system. In this study, entities are models of significant elements that exist in the real-world system; these include *student, task and teacher*. It should be noted that the human user of the system, the teacher, is also considered an entity. The process of identifying and selecting these entities will be explained later in this chapter.

Attribute — Refers to the variable or fixed properties (values) that define an entity, e.g. `STUDENT_MOOD` (variable) and `STUDENT_GENDER` (fixed). Values of both fixed and variable attributes may impact on an entity and the overall system state.

Abstraction (Reduction Simplification) — This is a summary of the key characteristics that define a system from a particular viewpoint, or “an analogical model of the target system” (Edmonds, 2000, p. 3). Abstraction can be thought of as the process of simplifying reality, “but only to the point

where there is no significant loss of accuracy” (Shannon, 1977, p. 35). Terán and Terán (2003) suggest that the process of “simplification” is essential for the usefulness and understanding of a system. Shannon (1977) offers a logical overview of the processes involved in the model formulation before abstraction may commence. First and foremost, it is essential to substantiate what purpose and/or problem the simulation will address. This requires a thorough understanding of the environment in which the system exists and operates. The next step is to identify all pertinent components that contribute to the operation of the system. System reduction may then occur through the disposal of components deemed insignificant to the simulation objectives. This is, Shannon (1977) says, often a difficult moment in the modelling process because a component’s significance is not always clear at this point in time. In the case of the VCS, component (entity) identification was relatively straightforward. Itemisation and reduction of entity attributes and the specification of the operational relationships between the entities and their attributes was, however, a more complex task.

Model/Modelling — An abstraction of the real thing (Gershenson, 2002), or what Terán (2004) describes as a representation of the abstraction itself. Kuipers (1986) uses the term “shallow model” in his characterisation of an expert system’s application domain. A complete system model normally consists of a collection of sub-models of objects and processes contained within the system. Terán (2004) makes a specific distinction in the process of modelling by defining two subgroups, namely *descriptive modelling* and *simulation modelling*; the former being a true high-level description of the system to be modelled, and the latter more inclined to be represented in a simulation language. This study shares this sentiment, although *conceptual modelling* is a preferred terminology over *descriptive modelling*. Section 3.3 provides some novel examples of the model and modelling process.

Simulation ‘To simulate’, as suggested by The Oxford English Dictionary (1989), may imply “to imitate the conditions or behaviour of (a situation or process) by means of a model, esp. for the purpose of study or of training”. A browse in literature (e.g. O’Keefe (1986); Shannon (1992); Banks (1999); Davidsson

(2002); Sargent (2005)) highlights some researchers' recurring use of the words *system*, *process* and *model* as essential elements of simulation. Some papers offer little or no attention to the various ways in which a simulation may be conducted, although, as Banks (1999) rightly points out, a simulation need not involve computer technology. Simulations are often subject to categorisation, such as field/context of study (e.g. social simulation and agent-based simulation), techniques (e.g. continuous and discrete event) and intended use (training, entertainment and observation). In this study, the VCS is considered to be a social simulation using various simulation techniques for the purpose of training.

3.2. Social Simulation

The nature of modelling for the social sciences, which Valente (2005) coins as 'qualitative simulation modelling', is arguably different from that of the natural sciences (Valente, 2005; Terán, 2004). According to Terán (2004), the involvement of cognition and decision making abilities in social simulations make this area more complex than traditional simulations such as industrial systems. The non-linearity of social phenomena present in fields such as the social, cognitive, behavioural and organisational sciences, Goldspink (2002) says, presents a challenge to existing simulation methods.

Aspects of human cognition and decision making abilities of VCS students imply complexities throughout all facets of this study; from conceptualising aims and objectives to modelling, implementation, use and evaluation. The overall challenge here is how to simulate something that is inherently unpredictable, and subsequently how to ascertain that the unpredictable was accurately simulated. Or, conversely, how can anyone contest that any (social) simulation output is incorrect? It is an impossible task, but, as this chapter will demonstrate, a few conditions and an approximation of *likely* behaviour output may still suffice in meeting the desired aims of the VCS.

3.2.1. Agent Based Social Simulation

Although the categorisation of the VCS as a social simulation is explicit, some properties of the simulation make it difficult to further elucidate where it sits in the terminology. In a recent article, Skrødal et al. (2007) described the VCS as an Agent Based Social Simulation (ABSS). The analogy was inspired by Davidsson (2002), whose paper on ABSS highlighted the intersection of three scientific fields; the social sciences, computer simulation and ABSS. Thus, ABSS can be seen as a subcategory of the broader realm of Agent Based Modelling (ABM)/Agent Based Systems (ABS)/Agent Based Modelling and Simulation (ABMS).

Skrødal et al. highlighted the ambiguity of exactly what constitutes an ‘agent’ and provided a definitive description of the ‘agent’ in their context:

In the VCS, an agent is a software implementation of a real-world human model, the student, which operates with its own set of beliefs, desires, intentions and emotions (BDIE). The agent model is based on a specific reconstruction of an individual (i.e. a student) derived from an expert deconstruction of what is considered to be essential attributes in a classroom learning context. Skrødal et al. (2007, p. 2)

Further research in the areas of ABSS/ABM/ABS/ABMS made it clear that little was gained in attempting to position the VCS in this terminology. Because a variety of definitions in literature were both inclusive and exclusive of the VCS’ fit, the decision was made to avoid any further references to this area.

3.3. What is Modelling?

At some point in the modelling process, a distinction must be made between *what* a system does and *how* it does it. Knowing how to operate a vehicle is not synonymous with knowing how a vehicle operates. A timepiece, for example, may serve as a decorative feature, a collectors’ item or a nostalgic keepsake. It’s primary function, however, is to tell the time — this is *what* it does. With regard to explaining *how* it does it, things get a little more complicated; not only because different technologies can achieve the same result (e.g. analog vs. digital), but also because

3.3. What is Modelling?

the explanation/description of a system may be delivered in an endless number of ways. Consider the watch; its function can be superficially described as being attributable to a particular configuration of mechanics. However, if studied in more detail, one would at some stage expect to learn more about how this configuration works, the purpose of cogs and so on. Understanding exactly how a clock tells the time could therefore imply learning about how a screw works, how a particular spring serves a specific purpose in the configuration or even how we have come to define time in the first place. And should theoretical explanations at any level not suffice, the practical approach of dismantling the device, as has been done with the analog wristwatch in Figure 3.2, might also be useful.

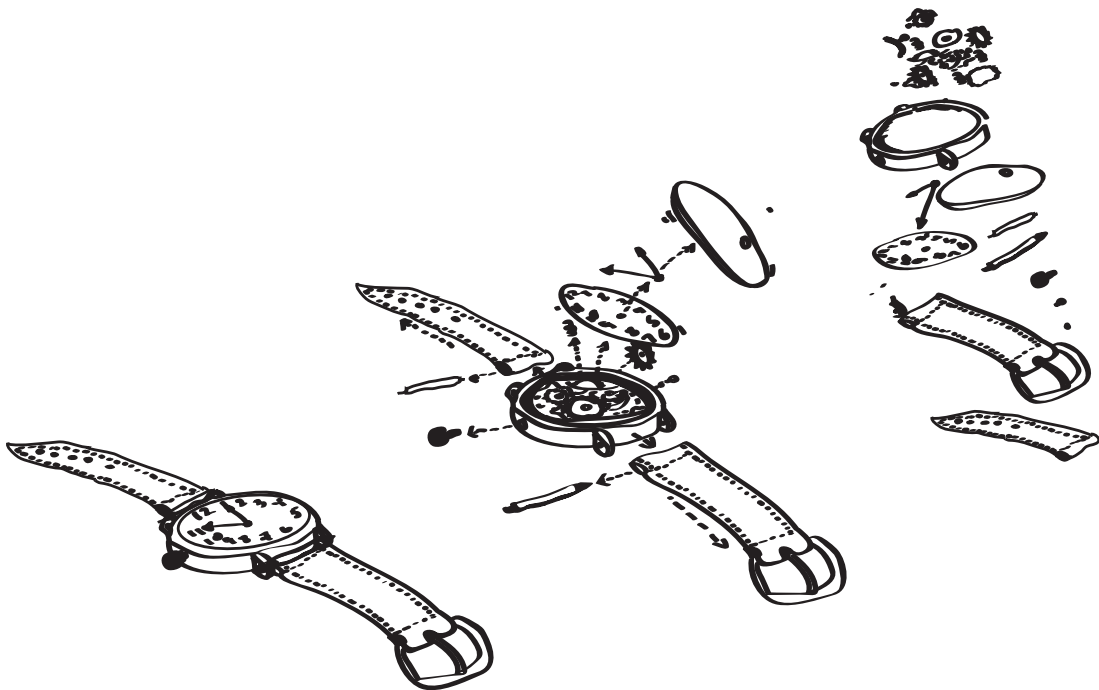


Figure 3.2.: System Deconstruction of an Analog Wristwatch

With all pieces of the watch identified, and a thorough understanding of how they work together to tell the time, one can begin creating a conceptual model. Shannon (1992) stresses that modellers too often tend to bloat their models with too much trivial detail, causing unnecessary complexities and the potential loss of significant aspects and relationships in the simulation. As such, Pareto's principle of

the significant few and the trivial many, or the 80/20 rule, is a suitable concept in this regard (Shannon, 1992). Edmonds (2005, p. 4) describes this as a necessary trade-off where “abstracting away from a descriptive realism achieve a greater generality”. Thus, if a model’s purpose was to illustrate how the mechanics of the wristwatch in the above example work, certain parts should already be excluded, such as the wristbands, front and back covers, the dial, and even its casing. In this instance, these components are not relevant to the inner mechanics, and would unnecessarily complicate the model if included. Although it is not always easy to distinguish the trivial parts of a system from the vital ones, early efforts in this area will be rewarded when formulating the simulation model and implementation.

Given the overall context of this thesis in particular, one may find it tempting to link theories and examples of modelling directly to their application in computer science. However, as shown with the example above, this is not always the case. Modelling processes are arguably part of everyday life, in our subconscious analysis of people, artefacts, social systems and so on. In fact, the construction of a model is, by and large, still operating within the realm of examining theoretical and practical aspects of a system. A model, which at the most basic sense can be thought of as a description or impression of a particular system (or parts thereof), may therefore serve as a basis for simulation, but the involvement of computer science is not implicit.

Consider, as an illustrative example in physics, a sub-system of gravity studied in isolation; the concept of free-fall and projectile motion. The observed system here is concerned with gravity’s effect on two objects, A and B, released as free-fall and projectile respectively. A real demonstration can of course be achieved simply by releasing two objects from the same height, one as free-fall and the other as projectile, and observing their time of flight. This particular system can also be modelled mathematically (e.g. $h = v_0t + \frac{1}{2}gt^2$, where h =height, v_0 =initial velocity, g = acceleration due to gravity and t =time of flight) which may be useful for simulation/mimicking purposes (e.g. sketches on a blackboard when teaching physics). Thus, the creation and use of a mathematical model (using a formalised language) as an enabler for the simulation of a system need not involve computer science either. Given a detailed model of the workings of a particular system, its implementation in a computer simulation should nonetheless be relatively

3.3. What is Modelling?

straightforward. However, as alluded to in Section 3.2, to model and simulate isolated systems governed by the laws of physics are arguably less complex and easier to validate than, say, systems involving the social sciences and human behaviour (Axelrod, 1997).

In the case of the VCS, the process of conceptually pulling a system apart and then piecing it back together is not as tangible and straightforward as with the wristwatch. In the latter system, the entities and their functional relationships with each other are known, or at least observable. In a social system like the classroom, there is no telling how many ‘parts’ are involved, let alone what they are and how they operate. Even when the main entities of the VCS system have been established, the fixed and variable attributes of these, and how they together form a network of interrelated processes, cannot be fully understood. While a mathematical function can accurately predict the time of flight of an object, there is no equivalent function to precisely predict human behaviour.

Arguably, our informal and often subconscious analysis and understanding of everyday phenomena involve continuous processes of deconstruction. Human beings are quite good at piecing together fragments of conceptual knowledge and past experiences to improve the validity of our perceptions. This is also true for conscious and formalised deconstruction processes, which Dix (2003) suggests are essential in science and academic study. The process of practically or conceptually taking a system apart and then piecing it back together relates very closely to deduction, in that it generates a set of precise assumptions about a system in terms of what and how it works. Simulation, Axelrod (1997) says, is a third way of doing science which involves both induction and deduction.

In bridging the gap between a conceptual and a simulation model, Shannon (1977, p. 34) emphasises four distinct tasks for converting a system into a set of algorithms:

1. Specification of the purpose of the model;
2. Specification of the components to be included in the model;
3. Specification of the parameters and variables associated with the components;
4. Specification of the functional relationships among the components, parameters, and variables.

In the case of the VCS, the purpose of the model reflects the aims and objectives of the study covered in previous chapters. The remaining tasks are detailed later in this chapter.

3.4. VCS Conceptual Model (VCS Entities)

This section and the next address the final three of Shannon's four tasks listed above, namely the specification of the components to be included in the model, the parameters and variables associated with the components, and the functional relationships among the components, parameters, and variables.

In order to mimic a system, in this case student-teacher interaction in a classroom environment, entities deemed as significant parts of the system must be identified (Nordgren, 1995). Further examination should then in turn isolate any attributes associated to these entities and ascertain how, and to what extent, they impact on each other. Only when the system has been methodically *deconstructed* can a model be pieced together, keeping in mind that the inherent complexities of the real-world system may render an exact replica impossible. This is certainly the case with regard to the VCS, where a complete *reconstruction* of the classroom environment is much too complex. Pareto's principle of the essential few and trivial many has therefore been a useful concept, where delimitations were enforced to ensure focus on the specific areas important to this research study.

Work on the VCS as part of the Honours study highlighted four significant entities of student-teacher interaction in the classroom (Skrødal, 2003):

- Class (see Table 3.1 on the facing page)
- Student (see Table 3.2 on page 46)
- Task (see Table 3.5 on page 54)
- Teacher (see Table 3.6 on page 60)

The tables referred to above provide a comprehensive catalogue of each entity's attributes, aspects and processes. These were produced by the two supervisors in education and the researcher. A number of educators not explicitly involved with

the project were also invited to review the lists and add further input and validation. The effort of ‘deconstructing’ the student entity is in itself a significant contribution towards understanding the multiple factors that affect student behaviour in the classroom setting. This work has been further advanced in the current study.

3.4.1. Class

The class entity involves a number of fixed and variable properties that describe the students as a group and the setting in which the lesson takes place. Table 3.1 highlights the class attributes, values and processes identified in the modelling work undertaken in the Honours study.

Table 3.1.: Original Class Attribute Listing (Honours Study, Skrødal, 2003)

Attribute	Values	Process
Atmosphere	Easy/Moderate/Difficult	Fixed 1,2,-1,3,-1
Lesson type	Regular/Tutorial/Group/ Lab/Field Trip	Fixed 1,2,-1,3,4,5,-1
Year level	Primary/Middle/Secondary	Fixed 1,2,-1,3,-1
Class size	Small/Large	Fixed 1,2,-1
Time of lesson	First Lesson/Before Recess/ After Recess/Early Afternoon/ Last Lesson	Fixed 1,2,-1,3,4,5,-1
Length of lesson	Regular/Long	Fixed 1,2,-1
Gender	Coed/Boys/Girls	Fixed 1,2,-1,3,-1
Weather	Normal/Unusual	Fixed 1,2,-1
Equipment	Functioning/Problem	Fixed 1,2,-1
Cognitive state	High—Low	Continuous 0-1
Affective state	High—Low	Continuous 0-1
Behaviour state	High—Low	Continuous 0-1

The list, which provides a subset of class attributes with potential relevance for the original simulation objectives, was delimited further in this study. Attributes

that are explicitly or implicitly implemented in the current version of the simulation include lesson type, year level, class size, length of lesson and gender. Only lesson length may currently be altered by the user (at the start of each simulation run); all other properties are fixed and cannot be changed. Extension in this area is one of many features considered for future versions.

The cognitive, affective and behaviour states are implicitly present in the current version of the VCS, that is, as an average of individual student values. The original intent for these attributes, however, was to implement them as an overall ‘classroom climate’ — another area for future work.

3.4.2. Student Entity

From an education perspective, the student entity is the most complex component to model in the entire VCS system. Since human behaviour is very circumstantial, it is also enormously complex and very unpredictable. Nonetheless, human beings appear to be quite good at interpreting and predicting how different modes of interaction will *typically* affect the behaviour of others. Since the VCS does not aim to simulate the entire spectrum of social interaction, but rather highlight specific traits in an isolated context, human interaction and behavioural models can be delimited as such. While the process of abstraction does offer some relief, it is still very difficult to separate the essential few from the trivial many. This has therefore been an ongoing and evolving task since the conceptualisation of the VCS.

It is important to realise that VCS students are essentially composed of a set of attributes (fixed and variable). The ‘Artificial Intelligence’ (AI) that may be conveyed by the simulation is the product of mathematical models’ continuous processing and updating of numerical data associated with these attributes. Because every student instance is based on the same model, they share identical attribute declarations. The variations and differences in attribute values is what makes each student unique. Once the student entity structure has been determined, the mathematical model can therefore cater for any number of instances, allowing for a flexible class size.

3.4.2.1. Earlier Work

An early assumption was that the VCS should mirror a class size typical for many schools in the real world, with the Honours version allowing 24 students in the class. Given the information above, this should not be a problem, since a properly constructed model can scale to 100 students just as readily as 24. Limitations in this regard may, however, be imposed by the host computer's memory and processing abilities. Moreover, the practicality of displaying a large number of students on a screen with limited real-estate became an issue. And there was no pedagogical reasoning for including 24 students in order to address the study's objectives. Following a review of early assumptions and expert advice from supervisors in education, a class size of 6 was found to be an ideal number for the purposes of the current research. Table 3.2 highlights the student attributes, aspects and processes identified in the modelling work undertaken in the Honours study.

3.4.2.2. Current Development

It is not within the scope of this research study to include all identified student attributes. Not only would it stray away from the simulation objectives, but the project could also run into the risk of losing track of the 'essential few' and render the system too complex. The task of specifying all parameters and variables associated with the student entity was nonetheless an essential exercise, as it provided a starting point from which further analysis and subsequent abstraction could begin.

In the current study, each attribute was carefully re-assessed for potential inclusion in the student model. The relevance of each attribute and how it would contribute to the operation of the simulation had to be considered on multiple levels. Identification of attributes significant to the research objectives, and how these can be modelled to a satisfactory level of accuracy, required a good insight into educational theory and cognitive science. The education course in which the VCS was to be integrated, 'Student-Teacher Interaction in the Classroom' (STIC), was therefore carefully investigated. In order to make the simulation relevant to STIC, educational theories underpinning the VCS had to be aligned with those taught in the course. Attribute identification was validated by expert advice from

Table 3.2.: Original Student Attribute Listing (Honours Study, Skrødal, 2003)

Attribute	Values	Process
Prior knowledge	Low/Medium/High	Fixed 1,2,-1,3,-1
Learning pref. (match)	Low/Medium/High	Fixed 1,2,-1,3,-1
Goal type	Performance/Learning	Fixed 1,2,-1
Distractibility	Low/Medium/High	Fixed 1,2,-1,3,-1
Disability	None/Minor/Major	Fixed 1,2,-1,3,-1
Disability type	Phys./Vis./Hear./Learn.	Fixed 1,2,-1,3,-1,4,
Age for class	Low/Avg/High	Fixed 1,2,-1,3,-1
Gender	Male/Female	Fixed 1,2,-1
Misconception	None/Weak/Strong	Fixed 1,2,-1,3,-1
Cognitive level	Low/Medium/High	Fixed 1,2,-1,3,-1
Required wait time	Short/Medium/Long	Fixed 1,2,-1,3,-1
Knowledge connectedness	High—Low	Continuous 0-1
Speed of processing	High—Low	Continuous 0-1
Specific strategy knowl.	High—Low	Continuous 0-1
Gen. prob.-solving knowl.	High—Low	Continuous 0-1
Metacognitive control level	High—Low	Continuous 0-1
Task self-efficacy	High—Low	Continuous 0-1
General self-efficacy	High—Low	Continuous 0-1
Attributional style	High—Low	Continuous 0-1
Emotional level	High—Low	Continuous 0-1
Persistence	High—Low	Continuous 0-1
Self-directedness	High—Low	Continuous 0-1
On-task	High—Low	Continuous 0-1
Disruption	High—Low	Continuous 0-1
Hands up	No/Yes	Fixed 1,2,-1
Performance	High—Low	Continuous 0-1

3.4. VCS Conceptual Model (VCS Entities)

supervisors and the course lecturers.

In the stage of making a final selection of attributes, the subject of accuracy also became a very important item on the agenda. Consultations with supervisors from both education and computer science had consistently highlighted that the accuracy of the VCS should not be the most crucial element of the simulation. While it is essential that the VCS expresses behavioural aspects common to those in real life, the simulation part is not as important as the training tool itself. Due to the considerable level of model abstraction, the VCS constructs a reality that cannot be observed in the real world. Besides, it would in any case be impossible to validate the accuracy of the cognitive and behavioural aspects of the students at an exact level. For training purposes, the simulation design aims to accentuate a number of specific aspects of student-teacher interaction, thus amplifying likely effects and consequences of the real world. For these reasons, ‘simulation accuracy’ is in this study a relative term which cannot be validated using mathematical models or strict mapping with real world observations. Rather, validation of the cognitive and behavioural output of the simulation becomes part of the education students’ learning experience. To discuss, reason and reflect as to why the simulation output may sometimes be ‘invalid’ in certain contexts can be just as, if not more, useful in challenging mental models as observing the real thing. For these activities and cognitive processes to occur it is imperative that, as stressed in Section 1.2 (Chapter 1), the VCS is not used as a stand-alone module. Course integration and pedagogical implementation, in terms of preparation, objectives, guided exercises, group discussions, assessment and similar, are needed to optimise the usefulness of the simulation.

Table 3.3 provides the final set of student attributes in the current study, along with example initial values for one VCS student. Each attribute is explained in detail below.

ID (`student_id`) Unique identifier for a student entity/object in database and software (primary key in database).

Name (`student_name`) The human-readable representation/name for a student. The student names (Aksel, Annelie, Elias, Kennet, Maja and Sunniva) were created using an online random name generator (<http://www.behindthename.com>) configured with a Norwegian bias.

Table 3.3.: VCS Student Attributes (Current Study)

Attribute	Value	Database Type	Source Code Type
ID (fixed)	1	INTEGER (PK)	int
Name (fixed)	Aksel	VARCHAR (10)	String
Age (fixed)	14	INTEGER	int
Gender (fixed)	Male	INTEGER	
Description (fixed)	Aksel is a dedicated student who works hard in most of his subjects. However, despite his confidence in subject matters and strong focus on future career opportunities, being an ESL student makes him a little shy and less confident with public speaking. Nonetheless, his involvement in lessons is still apparent, as he is always taking notes and listening with great concentration.	VARCHAR (700)	String
Wait Time (fixed)	7.0	DOUBLE	double
Self Efficacy (variable)	7.0	DOUBLE	double
Self Efficacy Threshold (fixed)	4.0	DOUBLE	double
Distractibility (variable)	3.0	DOUBLE	double
Distractibility Threshold (fixed)	6.0	DOUBLE	double

3.4. VCS Conceptual Model (VCS Entities)

Attribute	Value	Database	Source Code
Mood (variable)	4	INTEGER	int
Mood Threshold (fixed)	5.0	DOUBLE	double
Knowledge State (variable)	6.0	DOUBLE	double
Prior Knowledge (fixed)	6.0	DOUBLE	double
Metacognition (variable)	4.0	DOUBLE	double

Age (`student_age`) Implicitly defines the year level for the student. Aside from giving the user a perception of the student’s age, this attribute does not have any further implications in the simulation model.

Gender (`student_gender`) For user perception only; no further implications in the simulation model. The VCS has an even gender distribution.

Description (`student_description`) The description provides a limited insight into the personality and abilities of each student prior to the simulation run. This a significant aspect of the training tool, because all users enter the lesson with their own set of preconceptions and bias towards individual students and the class as a whole. The descriptions also predispose users to take into consideration factors that are actually not explicitly built into the simulation (e.g. family background, inter-student relationships). A useful post-simulation exercise is to reflect on the results and how they map to the user’s initial mental model about each student, e.g. “did the descriptions create a bias towards certain students during the lesson?” or “did the descriptions justify the students’ behaviour and cognitive abilities?”.

Student descriptions, which were created by the researcher in collaboration

with a school teacher and the education supervisory panel, aim to provide an approximate reflection of numerical attribute values. The numerical values of all student attributes are also accessible prior to the simulation run, although they are hidden by default. An option to view these values is provided because it caters for a more in-depth analysis and understanding of how the simulation works. It would be advisable for users to study these aspects of the simulation only after they have run a normal lesson.

Wait-Time (`student_wait_time`) Wait time, which is a fixed student attribute, is used in the mathematical model to compute number of seconds before a student will raise his/her hand. The computation takes this value along with various other attributes (e.g. task properties, knowledge and self-efficacy) into consideration when producing a final wait-time for each student.

Self Efficacy (`student_self_efficacy`) Self-efficacy may be defined as “beliefs in one’s capabilities to organize and execute the courses of action required to produce given attainments” (Bandura, 1997 in Sang et al., 2010, p. 2). In this study, self-efficacy is used to indicate a student’s confidence in being capable of solving a task. A student’s confidence (actual or portrayed) may be high despite a low level of knowledge, an important aspect of teaching that the VCS tries to convey. Confidence may increase/decrease in a number of ways. For example; if a student answers a question correctly students who knew the answer will gain confidence while those who did not may lose some. The mathematical model uses self-efficacy in conjunction with other student/task attributes to compute the likelihood of a student answering a question correctly. An attribute called “interest” was originally used here, but it was substituted with self-efficacy about half-way through the candidature following advice from supervisors in education. Both interest and self-efficacy fall into the same category of motivational skills (or will) in problem solving (Mayer, 1998).

Self Efficacy Threshold (`student_self_efficacy_threshold`) The self-efficacy threshold, a fixed attribute, influences the level of which the variable component changes throughout the simulation. The mathematical model

3.4. VCS Conceptual Model (VCS Entities)

takes this variable into account when calculating the degree of change in self-efficacy in the occurrence of an event (e.g. a student answers a question correctly).

Distractibility (student_distractibility) This variable indicates the attentiveness for each student. If distractibility is too high, the student is not in a position to solve a task and is unlikely to raise his/her hand. It is used in the mathematical model for this purpose, and values are continuously updated on a per-second basis. Distractibility levels increase if the teacher does not actively engage the students with new tasks.

The VCS has not yet reached a desired level of animation to provide visual cues about the students' level of involvement in the class (e.g. alertness, attitude and attention). It is nonetheless important to give the teacher some form of information in this area, so the current version implements a colour-coded distractibility meter for each student. The teacher needs to carefully observe these meters to ensure that all students are paying attention before a task is given. The user interface also provides means for the teacher to ask individual students to pay attention, causing the distractibility levels to drop for all students with the highest adjustment observed with the student addressed.

Distractibility Threshold (student_distractibility_threshold) Influences the level of which the variable component changes throughout the simulation. Some students are more easily distracted than others, a trait that is computed in the mathematical model using this property. Therefore, students also display different levels of adjustment to the variable component when asked by the teacher to pay attention.

Mood (student_mood) Involves two major components of an emotional behaviour system that Velásquez and Maes (1997) label as *expressive* and *experiential*. The expressive component entails the visual cues of student emotion, i.e. facial expressions. Six mood states are currently implemented in the VCS: “upset”, “sad”, “tired”, “normal”, “content” and “happy” (see Figure 3.3). These visual cues provide the user with an insight into the emotional state of each student.

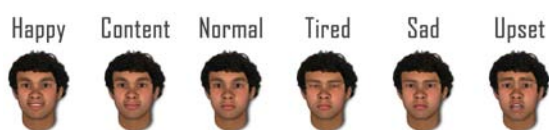


Figure 3.3.: Mood Range for Student Elias (see Appendix B for all students)

The experiential component, which is manifested in the mathematical model, is not yet fully implemented in the simulation. While it is functional and fully capable of demonstrating mood changes, the attribute is currently not taken into consideration in any other facet of the mathematical model (e.g. other aspects may affect mood, but not vice versa). A change in mood may occur in a number of ways, e.g. if a student without his/her hand raised is asked by the teacher to answer a question, if teacher does not give enough time to think before asking or if students have the right/wrong answer, to name a few. Counters keep track of the frequency of certain events that may eventually cause a mood change to occur. It is not known to the user what influences a student's emotional behaviour, but careful observation throughout a lesson run will provide clues. Moreover, mood is traced in the results graphs and can be coupled with event records from the database to establish exact cause for emotional changes.

Mood Threshold (`student_mood_threshold`) Influences the level of which the variable component changes on a per-event basis. Some students are easier to please (and upset) than others, and the mathematical model should take this threshold into account when calculating whether or not a student's mood state should be elevated or lowered. The mood threshold, which is fully integrated in the simulation framework, has yet to be taken into account when calculating student mood. While it would only be a matter of minutes to implement this in the software, the conceptual understanding was not quite right. All students therefore share the same threshold for mood changes in the current version. A more complex emotional behaviour system is therefore an important feature for future development.

Knowledge State (`student_knowledge_state`) This is a variable attribute that

3.4. VCS Conceptual Model (VCS Entities)

indicates how much knowledge a student has accumulated in a lesson. Its value is only visible in the results charts, which is typically examined at the end of the lesson (although the system does allow for the display of results during the simulation as well). Knowledge uptake can occur in a number of ways, for example: a student answers a question correctly (all attentive students who did not know the answer will learn); a wrong answer is addressed by the teacher (all attentive students who did not know the answer will learn).

Prior Knowledge (`student_prior_knowledge`) This fixed attribute suggests the level of background knowledge for each student. The higher a student's prior knowledge, the more advanced tasks he/she is able to solve. The mathematical model also takes into account other attributes when calculating the likelihood of a student answering a question correctly (e.g. task properties, allowed time to process, metacognition and self-efficacy).

Metacognition (`student_metacognition`) An attribute called "cognitive level" was originally used here, but it was substituted with metacognition about half-way through the candidature. As Mayer (1998) points out, there are significant differences between cognitive and metacognitive knowledge. Metacognitive knowledge (or metaskills), Mayer (1998, p. 53) says, "involves knowledge of when to use, how to coordinate, and how to monitor various skills in problem solving". These problem solving strategies separate themselves from the domain-specific skills and knowledge implied by cognitive knowledge.

The above attributes, and how they operate in the VCS, will be explored further in this and the next chapter.

3.4.3. Task Entity

The task is the most significant component in the context of teacher-student interaction. It is the user's continuous formulation and communication of questions (events) that drive the progression of simulation states and, as a consequence, encourage an adaptive mode of teaching. The multiple implications of the interaction between a task and the student implies that none of these

Table 3.5.: Original Task Attribute Listing (Honours Study, Skrødal, 2003)

Attribute	Values	Process
Question sequence	1st, 2nd, 3rd, 4th	Fixed 1,2,-1,3,-1,4
Question level	Low/Medium/High	Fixed 1,2,-1,3,-1
Question type	Recall, Elaboration, Analysis, Synthesis, Explanation, Evaluation	Fixed 1,2,-1,3,4,5,-1
Cognitive Load	Low/Medium/High	Fixed 1,2,-1,3,-1
Task Difficulty	Low/Medium/High	Fixed 1,2,-1,3,-1
Novelty	Low/Medium/High	Fixed 1,2,-1,3,-1
Independence	Low/Medium/High	Fixed 1,2,-1,3,-1
Divergence	Low/High	Fixed 1,2,-1
Task number	1st, 2nd, 3rd, 4th, 5th	Fixed 1,2,-1,3,4,5,-1
Task type	Introduction, Individual seatwork, Small group, Large group Q&A, Review, Homework	Fixed 1,2,-1,3,4,5,-1

entities can be solely modelled in isolation. A carefully forethought fit between the chosen attributes and processes of the task, and those of the student, must be in place in order for the interaction system to function.

3.4.3.1. Earlier Work

The task entity has, for many reasons, been a challenging one to model. While the Q&A interaction format employed by the current version of the VCS is in alignment with that of previous work, the task entity itself has undergone a major overhaul. An initial approach for establishing how a task should operate in the VCS involved a thorough analysis of the different modes of student-teacher interaction. Table 3.5 highlights some of these modes as well as other attributes, aspects and processes identified as part of the Honours study. The attributes chosen for the current study will be discussed in more detail later in this chapter.

3.4. VCS Conceptual Model (VCS Entities)

Prior to the current study, the assumption was that the VCS should be able to cater for any given subject area and year level. The Honours study, for example, provided two sample subject areas. Each area provided a small set of ‘real’ questions (tasks) with attributes calibrated for the subject’s designated year level. An advantage of this approach is that the VCS can allow users to teach a subject area and an age group of their preference and/or expertise. A modular design allowed for the construction of new subject data files (in the XML format) that could be plugged into the simulation, allowing practically anyone to create content for the simulation. It could be argued that this flexibility further lines the simulation up to the real system it aims to mimic. There are, on the other hand, a number of disadvantages to this method as well. Complicating the task entity will inevitably impact on the complexity of the student entity and the overall simulation framework:

- Practically, a single subject area needs a subject database of a significant size. Failing to provide the user with a sufficient range of questions, with a variety of attribute combinations, would limit the interactivity and simulation outcome. A good size question bank is also essential so as to avoid repetitive use of the same set of questions.
- Pedagogically, the calibration of a question’s attribute values is very tricky. To illustrate, a sample question for a Year 9 Information Technology subject in the Honours version of the VCS reads: “Why do we need passwords?”. Even if the task had only one attribute, e.g. ‘complexity’, how would one give a numeric value to this question’s attribute? Complexity compared to other questions in the database — how to differentiate? Complexity with regard to students’ attributes — how to determine? Complexity in terms of how much detail is required by the answer provided — implies a need for further meta-data. Design and validation of a pedagogically sound question database would arguably constitute a research study of its own.

3.4.3.2. Current Development

The researcher entered the PhD candidature with some firm assumptions about the task entity. Shaped by earlier work, these preconceptions were in many ways

hindering the progress of development in this, as well as other areas. Many structural and conceptual approaches were considered in order to practically and pedagogically address the abovementioned complexities. In the process of revisiting relevant literature and research objectives, however, it became clear that the task entity had been overcomplicated in earlier work. The objectives of the VCS as a training tool for preservice teachers never implied the need for specific subject areas. This was just a presumption made by real-world observation and the absence of a better alternative. Well into the modelling process, however, a researcher proposed a new approach. The suggested solution was to do away with subject area and subject-specific questions altogether. The VCS would no longer be concerned with what subject matter is being taught, which is a genuine supposition since the subject matter is ultimately irrelevant to what is being measured. One potential draw-back to this solution, however, is that it takes some of the ‘realness’ away from the interaction context.

As shown in Figure 3.4, the idea was to create categories of generic questions, ranging from Easy—Medium—Difficult.

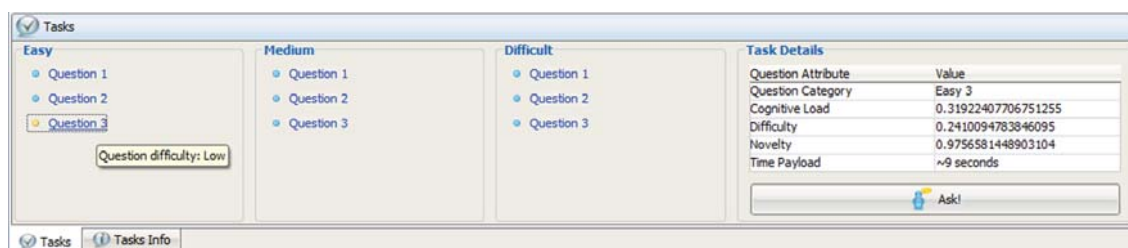


Figure 3.4.: First Version of Generic Tasks

Each category would have sub-categories with only minor changes to task attribute values, for example:

- Easy
 - Question 1:
 - * Random difficulty range between 0—0.1
 - * Random novelty range between 0.6—1
- Medium

3.4. VCS Conceptual Model (VCS Entities)

- Question 1:
 - * Random difficulty range between 0.3—0.5
 - * Random novelty range between 0.3—0.6
- Difficult
 - Question 1:
 - * Random difficulty range between 0.6—0.8
 - * Random novelty range between 0.1—0.5

Thus, an easy question is less difficult and more novel than a difficult task (which is likely to be more complex and less novel). The time it takes to ask an easy question (time payload) is also typically less than that of a more difficult one. The task details panel provides information about the task attribute values and lists the different random value combinations calculated each time a question item is clicked.

This alternative implementation was well received by the supervisory panel and the supervisors in education agreed that this solution was better suited for addressing the simulation objectives. Using this approach, the developer no longer needs to worry about creating and validating a question bank and the user is no longer concerned with what the question/subject is — only its attribute loading. The design does, however, breach an unofficial rule of the simulation interface in that it provides more detail than what would be observable in the real classroom. No teacher in the real world would know the difficulty of a question down to an accuracy of 16 decimal points! In consultation with Paul Calder, supervisor in computer science, an alternative approach was devised to uncomplicate and hide the unrealistic level of detail provided in the task details panel. The screenshot in Figure 3.5 demonstrates the final implementation of the task entity interface.

It consists of an interactive component (sliders) that allows the user to set approximate values for a question's difficulty and novelty. A visual component (coloured bars) is updated in real time to reflect difficulty and time payload (approximate time to ask a question) caused by the user's input. No explicit details about the question attributes are now shared with the user. Each time a question is asked, the simulation model will compute random values that fall



Figure 3.5.: Final Version of Generic Tasks

relatively closely within the ranges set by the user (e.g. two questions with identical settings will not necessarily compute identical difficulty, novelty and time payload values). An increase of both difficulty and/or novelty will amplify the time payload, though novelty carries the highest impact. The reason for this is that in order to make a question more novel/interesting, the teacher would typically need to invest more time in elaborating the task so as to motivate students.

The time payload has a minimum value of 1 and a maximum of 10, and is determined as follows:

- On the difficulty scale, time payload is divided into 1/4, i.e. a difficulty value of 65 gives a 2-point time payload.
- On the novelty scale, time payload is divided into 1/8, i.e. a novelty value of 65 gives a 5-point time payload.

In the above example, the combined time payload is $2 + 5 = 7$ points (out of 10). The time payload in actual seconds is not given to the user until the question is asked. The 10-point scale is only an approximate visual indication. In the simulation, 1 point might equate to approx. 4 seconds (e.g. a short yes/no question) while 10-points might equate to approximate 40 seconds (e.g. longer introduction/explanation of concept before asking the question). A 7-point question is therefore likely to take approximately 28 seconds on average. However, a random 'fuzzy' algorithm may add or subtract a few seconds from this average. In real life you never know exactly how long it will take to ask a question, although an estimate can be made and this is what the VCS model aims to reflect here.

It is not possible to compose a question with both difficulty and novelty set to maximum levels. When a question's difficulty becomes too high, the novelty level

3.4. VCS Conceptual Model (VCS Entities)

will automatically decrease (more on this in Section 4.2.3.7, Chapter 4). A question difficulty (D) above 75% cannot achieve a novelty value (N) of 100%. Where D is 76%, max N is 99%, where D is 77%, max N is 98% and so on until D is 100% and maximum value of N is 75%. This feature is based on the premise that there is a limit to how interesting a student can perceive a really difficult question. For future versions it is planned to make the task entity more sophisticated, de-couple D and N and let the virtual students process the thresholds based on their own attributes.

3.4.4. Teacher Entity

Most aspects of the teacher entity, which is *the human user* of the VCS, are external to the software implementation. Hence, whilst the behaviour of the teacher is certainly affected by the VCS and vice-versa, the teacher entity cannot be implemented in the same way as other entities integral to the system. Even so, it is important to be aware that the teacher entity is as much a part of the simulation as any other entity in the classroom. Inclusion of the teacher entity is achieved by providing the user with an interface for communicating with the system (e.g. facilitating for input and output functionalities). Although this communication interface is restricted according to the delimitations of the system, it still provides an environment where countless combinations and configurations of teacher-student interaction and behaviour can occur.

For future versions of the VCS, it has also been considered to include ‘soft’ attributes for the teacher. For example, a user might be able to adjust system attributes describing his/her personal traits, knowledge, skills, etc. as perceived by individual students. Although the virtual students’ attitudes towards the teacher are not considered in the current version, this feature would add another dimension to the simulation experience. Table 3.6 includes a number of teacher attributes, aspects and processes that were identified in work undertaken as part of the Honours study. Note that several attributes in this list are implicitly present in a VCS user (e.g. everyone has an age, knowledge, behavioural traits, etc.). The abovementioned ‘soft’ attributes would imply the programmatical inclusion of one or more of these attributes to impose the acknowledgment of teacher properties in the simulation model.

Table 3.6.: Original Teacher Attribute Listing (Honours Study, Skrødal, 2003)

Attribute	Values	Process
Age	Young, Middle Aged, Old	Fixed 1,2,-1,3,-1
Experience	Low/High	Fixed 1,2
Gender	Female/Male	Fixed 1,2,-1
Disposition	Positive/Cold	Fixed 1,2,-1
Pedagogical content knowledge	High/Low	Fixed 1,2,-1
Content knowledge	High/Medium/Low	Fixed 1,2,-1,3,-1
General teaching knowledge	High/Medium/Low	Fixed 1,2,-1,3,-1
Teaching style	Laissez-faire, Democratic, Autocratic	Fixed 1,2,-1,3,-1
Lesson preparation	Poor/Good	Fixed 1,2,-1
Quality of lesson introduction	Poor/Good	Fixed 1,2,-1
Teacher knowledge of class knowledge state	Poor/Good	Fixed 1,2,-1
Class coverage	One section, Whole class	Fixed 1,2,-1
Scaffolding	Low/Medium/High	Fixed 1,2,-1,3,-1
Presentation variety	Low/High	Fixed 1,2,-1
Behaviour management	Poor/Good	Fixed 1,2,-1
Question wait time	Short/Reasonable/Long	Fixed 1,2,-1,3,-1
Authority	Low/High	Fixed 1,2,-1
Feedback Information	Low/High	Fixed 1,2,-1
Affective feedback	Negative/Neutral/Positive	Fixed 1,2,-1,3,-1
Affective state	High—Low	Continuous 0-1
Behaviour	High—Low	Continuous 0-1
Withitness	High—Low	Continuous 0-1
Evaluation of student response	High—Low	Continuous 0-1
Teacher expectation of class	High/Low	Fixed 1,2,-1
Clarity	High—Low	Continuous 0-1

3.4.5. States of Interaction

Any form of social interaction can be thought of as being composed of one or more states (or modes). Interaction in the classroom, for example, typically involves a number of states in the teaching and learning context. Consider the act of the teacher asking a question to the class as one specific state of interaction. The processes and properties involved for all entities in this particular state (e.g. teacher, students and task) are significantly different to those involved when, say, a student is answering a question. The VCS recognises three unique states in the course of the simulation:

1. IDLE
2. QUESTION
3. INTERACTION

These states reflect the question and answering mode of teaching that the simulation aims to mimic. Thus, for each of these simulation states, different processes are activated in the simulation model as described below. Only one state can be active at any given time and there is always one state active throughout the simulation run.

As shown in Figure 3.6, the interaction states imply a modular design in the implementation of the simulation model. This makes it easier to programmatically monitor and control processes that may need to occur in certain time periods only. Occurrences and durations of interaction states are logged and later utilised in the final reports presented to the user.

IDLE State The IDLE state is activated at the commencement of the simulation. It signifies the time period where there is no Q&A-related interaction happening, e.g. when the teacher is preparing a new question for the class. Whilst in this mode, students are more prone to become distracted and this is the only state in which the teacher can ask students to pay attention. As soon as a question is asked by the teacher, the IDLE state is deactivated and the QUESTION state becomes active.

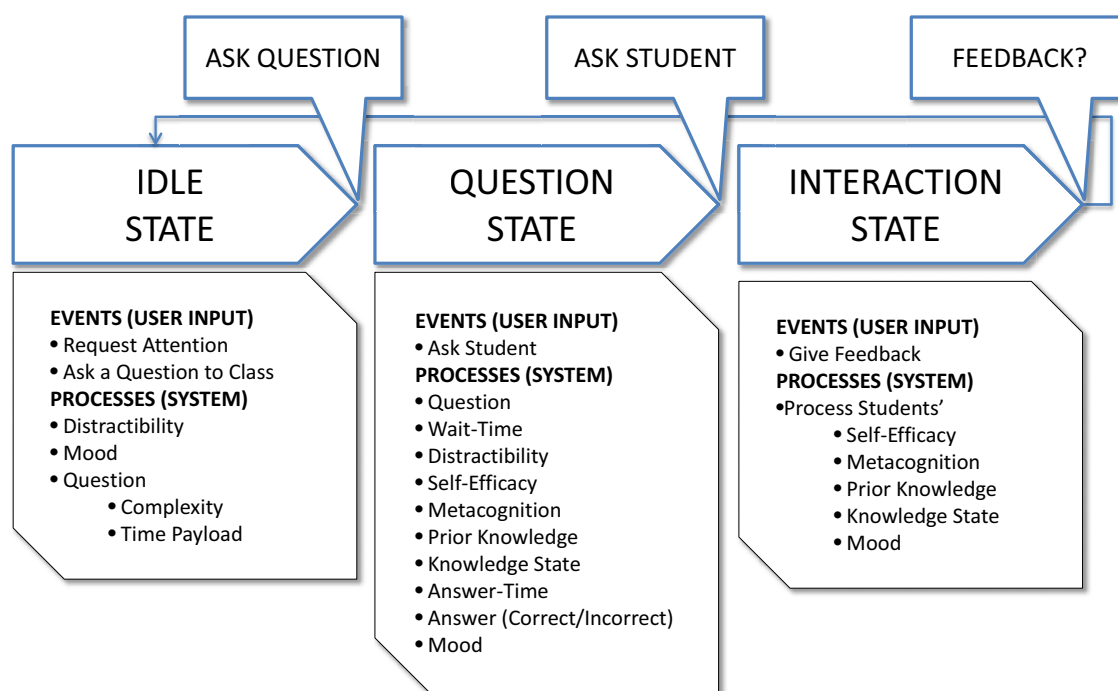


Figure 3.6.: VCS States of Interaction Overview

QUESTION State The QUESTION state runs from the moment the teacher asks a question until a student is asked to give an answer. Thus, the duration of this state depends on the time payload for asking the question and the period of time it takes for the teacher to select a student to answer it. At this point, the QUESTION state terminates and the INTERACTION state is activated.

INTERACTION State The INTERACTION state is active for the duration of the student answering a question. Another type of interaction that may occur in this state is the teacher providing feedback to an incorrect answer. The state terminates when a student has completed an answer or when the teacher's feedback has concluded. The IDLE state is then reactivated and a new loop of the three states commences.

The screenshot provided in Figure 3.7 illustrates how the states of interaction captured in the simulation database have been utilised to dynamically build the background of the results charts. Each interaction state has a label and a different

background colour. This extra layer of detail delivers information supplementary to simulation events and student attribute values.

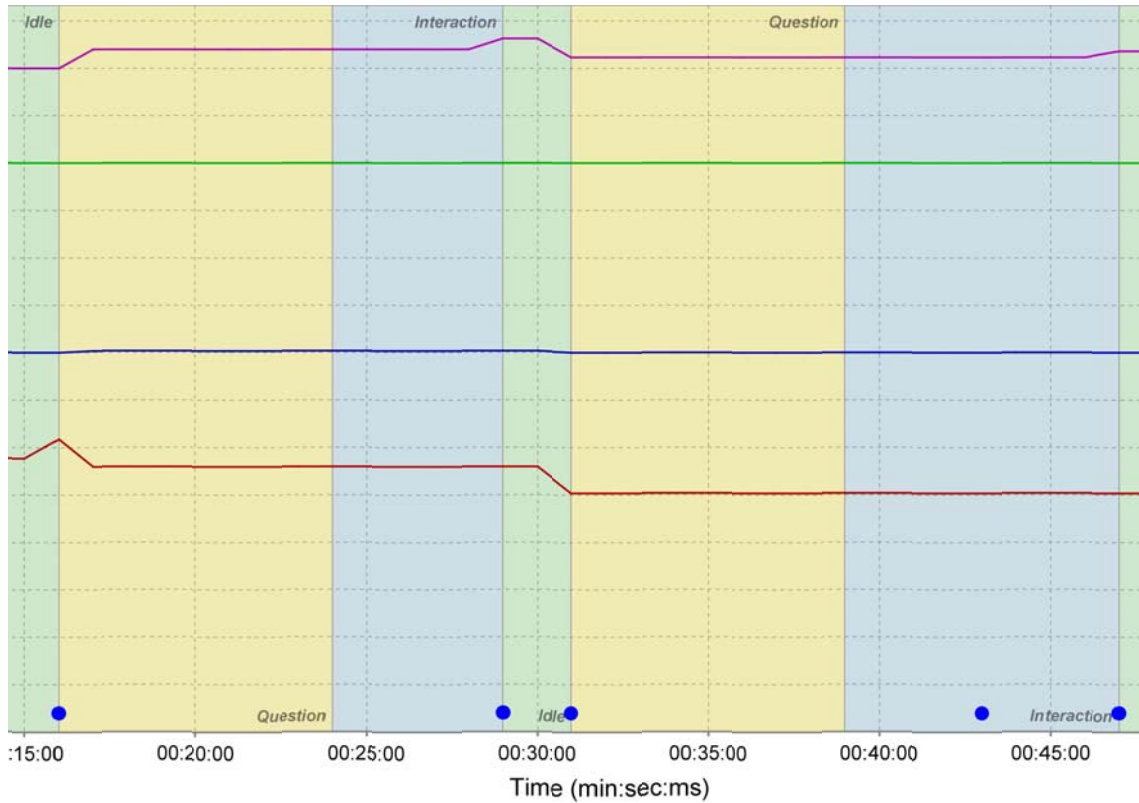


Figure 3.7.: Inclusion of Interaction States in Results Charts

3.4.5.1. Simulation 'Sweeps'

The continuous aspect of the simulation is defined in the simulation model through the use of 'sweeps'. Sweeps are quite simply regular updates (processing) and storage (output) of simulation data. The sweeps occur on a per-second basis, where the execution of processes is dictated by the current state of interaction.

3.5. Model Verification and Validation

A number of definitions for model verification and validation exist in literature. Nordgren (1995, p. 72) provides a straightforward understanding on the subject:

Chapter 3. Conceptual and Simulation Modelling

Model verification ensures that you have input data correctly, and model validation ensures that the model has successfully captured the operational characteristics of the system.”

In his terminology of model credibility, Schlesinger (1979, p. 104) defines model verification as the “substantiation that a computerized model represents a conceptual model within specified limits of accuracy” and model validation as the “substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model”. Sargent (2005, p. 135) says that conceptual model validation should determine “if the appropriate detail and aggregate relationships have been used for the model’s intended purpose, and if appropriate structure, logic, and mathematical and causal relationships have been used”.

Verification and validation of the VCS model has been an ongoing process throughout the development, implementation and trials of the simulation. As part of the verification process, mathematical algorithms in the model were singled out and stringently tested in the Microsoft Excel software prior to their inclusion in the computerised model. Because the real system that the VCS aims to represent is too complex to be modelled in its entirety, a number of delimitations were enforced. While these abstractions have made the model more manageable, there are still cognitive and behavioural aspects involved that are impossible to validate against the real system. The model may, however, be validated against its intended purpose; in situations where a model cannot be validated against the actual system, Nordgren (1995) recommends that it is validated according to the system assumptions. Similar to what is advocated by Sargent (2005), the VCS model validation was done using face validation (employing expert educational theory as well as feedback from user trials) and traces (tracking of entities to ensure correct logic and that accuracy is maintained).

“Just as validity is the sine quo non of theory, it is the ultimate and overriding requisite for simulations used for research and policy guidance. In educational simulations, however, it is not so much validity but credibility that is the primary requisite.” (Elder, 1973, p. 338)

3.6. Summary

Conceptual modelling efforts prior to this candidature produced a substantial amount of information useful in the current study. This information was primarily a product of entity deconstruction, e.g. a comprehensive breakdown of attributes relevant to classroom interaction. Earlier work did not, however, establish any conceptual or simulation models of interaction.

One key argument of this chapter has been the importance of abstraction in the simulation model. Elder (1973, p. 337) made it clear that “as the scope and complexity of the problem increase, focus becomes more difficult, crucial variables harder to identify, relationships blurred, and our knowledge more sparse”. Social simulations, particularly those that aim to simulate human cognition, are very complex from the onset. Reduction of the real system is therefore not only recommended, but inherently required.

Mayer (1998) highlights three essential abilities required for successful problem solving; cognitive, metacognitive, and motivational skills. All three abilities were included as attributes for the student entity in the form of prior knowledge, metacognition and self efficacy. In addition to these essential abilities, a VCS student entity also takes the behavioural aspects of distractibility, wait time and mood into account. The students’ problem solving abilities and behavioural aspects, along with interaction with other entities in the system, are used in the simulation model to promote a synergy akin to that in a real classroom.

The knowledge generated from the modelling phase provided essential guidelines and requirements for the simulation framework. The next chapter will discuss the particulars of framework development and how it was designed to address the model as well as facilitate interaction with the user interface.

Chapter 3. Conceptual and Simulation Modelling

Chapter 4.

Simulation Framework

The programming of a model is preceded by program design, which is subsumed by model design, which is a conceptually far more complex undertaking

HENRIKSEN (1983, p. 1053)

Previous chapters have established a foundation upon which the computer programming development and design can be based. The formulation of a conceptual and simulation model, in particular, was essential in providing a set of guidelines and requirements needed to operationalise the simulation framework. Conceptual and simulation models do not, however, stipulate *how* this is to be achieved. A number of considerations must be made in order to identify, develop and piece together modules and technologies that collectively address the requirements of the model as well as the second research problem (RP2) of this study:

Problem 2. Identify appropriate technologies (hardware and software) that combined allow for efficient and dynamic implementation, management and compatibility of the VCS with regard to:

- input/output (I/O)
- graphical user interface (GUI)
- stability

- speed
- usability
- extensibility
- reliability
- functionality.

While research and development in earlier work produced a few valuable resources for the current study, only limited progress was made in the conceptual and simulation modelling. Without a solid understanding of the model design, the construction of a simulation framework is troublesome. In the current research efforts, however, the unison of theory, modelling and computer simulation has been paramount. A clear and early conception of VCS features, functionality and underlying models have enabled parallel work in all areas, as well as cross-fertilisation of new ideas. While Henriksen is not incorrect in his introductory quote of this chapter, the many complexities of interface, framework and model programming should, however, not be understated.

Considering the scope and magnitude of this study, it was not possible, nor feasible, for the VCS software to capture every facet of the theory, research and development discussed. As the sole developer of the entire software system, the researcher had to make design decisions guided by the skills and knowledge at hand, expert advice, research evidence, experiences in commercial software development principles and time and resources available. These limitations did, of course, cause some discrepancy between what was desired and what could realistically be achieved. Nonetheless, features and functionalities incorporated were carefully selected so as to enable a substantial system capable of addressing relevant aspects of RP2 of this study.

This chapter provides an apt recount of the particulars of software design, as well as the resolution for problems and challenges related to the planning, implementation and deployment of the VCS.

4.1. Earlier Work

The aims and objectives of the simulation were relatively well defined even at the early stages of conceptualisation. The speculation was that an interactive computer simulation of student-teacher interaction could be useful in reshaping education students' mental models and in highlighting specific aspects of the educational theory. Exactly how the VCS would achieve these objectives, however, was not as clear. While early discussions around the VCS had not yet reached that level of conceptual and technical granularity, 'rapid prototyping' became a very useful exercise for conceiving many of the requirements and features applied in the current version (see Chapter 5).

Prior to the Honours candidature, the researcher's knowledge about computer simulations did not extend much further than to arcade games such as car and flight simulators, and the area of conceptual/simulation modelling was a foreign territory altogether. Nor was the researcher, a computer scientist, particularly specialised in education. Thus, initial emphasis focused on learning more about the educational theory underlying student-teacher interaction and areas of which that would be most suitable for simulation inclusion in order to address the key objectives. Meeting with expert educators was therefore useful in adding to the knowledge base, as well as refining and cultivating new ideas for the system. Similarly, expert advice from the supervisor in computer science was instrumental in getting the project up and running in an official capacity as part of an Honours candidature.

4.1.1. Honours Work

The advent of the Honours candidature afforded more resources to be put into the research and development on the VCS, with most focus drawn towards what could be achieved with regard to simulation complexity and framework implementation. That being said, other aspects of the simulation, such as the educational theory, modelling (discussed in Chapter 3), graphical user interface (GUI, discussed in Chapter 5) and data structures (discussed in this chapter), were also worked on in parallel.

4.1.1.1. Simulation Framework

The Honours candidature had only just commenced when a contact external to the university recommended a particular framework thought to be suitable for the VCS. Developed over a number of years at the Argonne National Laboratory (ANL), the Dynamic Information Architecture System (DIAS) is a Java-based object-oriented framework for modelling and simulation. An ANL representative produced an impressive back-catalogue from several governmental (USA) projects utilising the DIAS framework and suggested that it would be suitable for the VCS. Although the demonstrated projects were conceptually and functionally very different from how the VCS was envisaged to work, a limited understanding of framework requirements and no other alternatives led to the decision to purchase a license.

From the onset DIAS presented a number of challenges. Delivery of the framework was delayed by several months and only made available half-way through the candidature (Skrødal, 2003, p. 38) and the documentation was incomplete, often making references to sections that had not yet been written. Although the framework was closed source, some source code for sample projects was included. However, these samples lacked adequate comments and were also at times erroneous. And, since the framework had been under development since 1993, many features of the implementation were using custom formats and solutions that were not up to date with current standards.

Although the study was somewhat successful in implementing parts of the framework (e.g. simulation models, processes and initialisation), it never produced a working version. In particular, it was difficult to integrate the discrete event nature of DIAS with the continuous and discrete event requirements of the VCS. A long-distance phone conversation with one of the main developers of DIAS in the USA added further credibility to the difficulties experienced. She confirmed that DIAS did not, nor was it designed to, facilitate features required by the VCS. Having spent considerable amounts of time and effort to integrate DIAS with the VCS, this message came both as a disappointment and relief. All things considered, however, as an initial framework for exploratory development, DIAS provided an opportunity to learn more about simulation development and gain a

deeper understanding of the various aspects in need of consideration. In other projects where it had been successfully implemented, DIAS undoubtedly added much value by providing a solid framework which would have eased the development process.

The final decision to discontinue the use of DIAS in future versions of the VCS was made in 2005, 18 months after the completion of the Honours candidature. The researcher was sponsored to travel to Brisbane (Australia) for a 3-day course on simulation development using DIAS. This course did not provide any new knowledge with regard to DIAS implementation and the presenters experienced difficulties similar to those of the researcher in following the system's documentation. At the end of the course, they had been unable to piece together a working demonstration of DIAS, and the hands-on workshop did not eventuate. As a consequence, most of the subject matter covered in the course revolved around basic concepts of programming in Java. The experience confirmed that further efforts using the framework was unsuitable for the VCS and that DIAS at that point was insufficiently intuitive and lacked proper documentation and examples.

4.1.1.2. Data Input/Output (Data Management)

All computer software and hardware (or indeed computing components) require the ability to send (output) and receive (input) information (Wolf, 2001). Processes of data Input and Output (I/O) exist on many different levels and are achieved in a number of ways. The keyboard, for example, takes physical human interaction (output) as input. Information captured from the keyboard is digitised and sent (output) to the computer which perceives that data as input. The computer may instruct that this information be sent to a monitor (received as input) and displayed (output).

Luckily for the computer programmer, operating systems and programming language features commonly provide higher-level services that ease I/O operations through levels of abstraction. In practical terms, this means that the programmer need not be concerned about the lower-level workings of peripheral devices for I/O and hardware interaction (e.g. CPU and memory addressing). Higher-level I/O processes that occur within the simulation system, however, are in need of careful

attention from the developer. The actual structuring and handling of program-specific information is often subject to a great deal of attention, as it typically affects system design and performance. This is true for the VCS, where the bidirectional flow of data is more or less constant. Moreover, most of this information need to be stored (input) and subsequently presented (output) in a human-readable format. Throughout the development of the VCS, management and organisation of information has gone through several stages of improvement. In order to better understand the design features and technologies for data management employed in the current version of the VCS, it is therefore useful to first review some of the requirements, issues and solutions of I/O implementation in the Honours candidature.

It was clear that the interactive nature of the simulation would demand that the software be able to receive input from the user via peripherals such as the mouse and keyboard. Input would also be received from the VCS' own configuration and simulation data, data in files external to the program, as well as information from the Operating System (OS). Accordingly, the VCS would also need to store back to the system its results, configuration and log files, as well as direct information and services to the monitor display. While there are many known methods and technologies available for data storage and retrieval, their features, complexity, overhead and suitability for use in different scenarios are normally circumstantial. It would, for example, be an overkill to embed a comprehensive Database Management System (DBMS) in an application whose only requirement for data storage is a small number of configuration items. Potential drawbacks in this scenario would be increased application size, start-up time, compatibility issues, complexity and cost. Finding a best solution to a problem can be difficult, and the variety of technologies, standards and formats can sometimes be more confusing than helpful. Further, external modules may leave the developer with no choice but to follow the approach for data management used by those modules — as was the case with the DIAS framework, which imposed its own custom file format for configuration and simulation data.

The desired solution for managing data in the VCS was always to employ a DBMS. Aside from efficient management of potentially large amounts of data (scalability), the Structured Query Language (SQL) would add greater flexibility

and ease in the storage and retrieval of data. Although some DBMSs were available in the early days of VCS development, specific requirements restricted the uptake of any of these. A DBMS suitable for the VCS had to be free, portable (should run on any system, just like the VCS), embeddable (no need for separate installation, downloads, drivers, etc.), interface well with Java and its libraries for DBMS connectivity and manipulation, and be relatively quick and simple to implement. The use of a DBMS in the Honours candidature was, however, not feasible. Much time and resources had already been lost due to delays and issues with DIAS. Moreover, DIAS imposed its own custom formats for data input and output which would have made the transition too complex. A decision was therefore made to find an interim solution at the time.

INI File Format The very first running prototypes of the VCS (developed in 2003) relied on the simple INI (initialisation) file format for data input (Skrødal, 2003). In some cases, programming and scripting languages may include libraries or extensions that already support particular formats ‘out of the box’, as was the case with the Java Application Programming Interface (API). Conveniently, the key=value data structure of an INI file is the same as that employed by the `java.util.Properties` Class (since JDK1.0). This made the I/O of basic data structures relatively simple to implement programmatically. Although Java simplifies the storage and retrieval of properties in this format, it was far from ideal for the structures and amount of data the VCS was envisaged to capture. The decision to use the INI format was mainly dictated by DIAS, which at the time used this format for initialisation data input as well as a custom plain text format for its data output. Such output structures present a problem when they need to be re-read back to the system as data input (e.g. to report results). Since they do not adhere to any particular standard, custom code and internal data structures/logic have to be devised in order to make any sense of the data. Moreover, if the output structures were to change (e.g. adding/altering output data structures), such custom solutions would also have to be altered accordingly. The aforementioned issues along with the limited nature of the INI format (particularly, it did not map well to the object oriented design of the VCS software) prompted a move towards an alternative data structure for the VCS.

XML File Format The Extensible Markup Language (XML) is an open standard for structuring information in a way that suits storage and handling of object-oriented data particularly well (see Quin, 2008). Uptake of XML was in 2003 on the rise, although not nearly as widespread as today where its many flavours of markup languages and areas of application are demonstrated in a vast array of solutions online, on mobile devices and in traditional desktop software. It was an appealing format for the VCS because it was portable (plain text), light-weight, allowed for more flexible structures of data representation, object-oriented and self-documenting. Further, the Java API for XML Processing (JAXP) included interfaces for parsing XML documents.

For these reasons, the VCS implementation moved away from the INI file format and adopted Java's Simple API for XML (SAX) interface for parsing XML data. The final XML structure design of a VCS class file, as partially shown in Algorithm 4.1, contained initial (start-up) parameters for the class as well as its students. A similar XML structure was also defined for each subject area available for use in the simulation with a `<details>` tag to describe the subject (e.g. title, year level, description, etc.) and a number of `<question>` tags (one for each question available for the teacher to ask the class). In 2004, the year after the completion of the Honours candidature, the researcher developed a class/subject editor with a GUI that allowed the user/administrator to create their own class/subject files with ease.

Because the integration of the simulation framework (DIAS) never reached a stage in which generation of simulation output would occur, no mechanism was developed for the storage of data. XML was therefore a temporary solution to the data format and storage problem. While XML was considered well suited for the storage and loading of start-up parameters, the implementation and structure of data output (as well as its subsequent interpretation and presentation) would require a major re-think. No further work was undertaken to address these issues in the Honours candidature. They were therefore labelled as "in need for future development" in the final write up (Skrødal, 2003).

As a side-note, in regard to the abovementioned file formats employed by DIAS, the researcher shared his XML experiences with one of the main developers of the framework at ANL. This information was very well received and it generated interest in the potential of the XML format for the DIAS framework. It is not known if these

Algorithm 4.1 XML Structure of a VCS Class

```

1 <?xml version="1.0"?>
2 <class>
3   <details>
4     <name>Demo Class 1</name>
5     <year>9</year>
6     <author>Simon Skrodal</author>
7     <date>03.09.03</date>
8     <description>
9       A sample year 9 class with no huge variations in
        attributes.
10    </description>
11  </details>
12
13  <student>
14    <name>Emma</name>
15    <gender>f</gender>
16    <race>w</race>
17    <id>2003005</id>
18    <maxinterest>10</maxinterest>
19    <mininterest>0</mininterest>
20    <interestlevel>6</interestlevel>
21    <interestthreshold></interestthreshold>
22    <metacognition></metacognition>
23    <raisehandinterest></raisehandinterest>
24    <raisehandknowledge></raisehandknowledge>
25    <handraised>0</handraised>
26    <relationshiplevel></relationshiplevel>
27    <knowledgelevel></knowledgelevel>
28    <answeredquestion>0</answeredquestion>
29    <ignoredcount>0</ignoredcount>
30  </student>
31
32  <student>
33    ...
34  </student>
35 </class>

```

suggestions and recommendations contributed to the uptake of XML in DIAS.

4.2. Current Development

Framework development in the current study has enjoyed benefits from experiences gained in earlier work. Previous efforts highlighted many of the difficulties involved and suggested a few alternative options for future development. The researcher's problem solving skills, programming experience and content knowledge were also improved in this process. Further value was also added by the availability and affordability of new/updated tools and resources for software development (e.g. the Java Development Kit, programming environments, third party modules, hardware and other software).

However, it was clear from the onset of the PhD candidature that the DIAS framework was not suitable. Moreover, no satisfactory solution, not even conceptually, had been established for I/O management. This meant that the simulation framework development up to this point constituted no tangible components suitable for reuse.

4.2.1. Simulation Framework

Although DIAS was no longer considered feasible for the VCS, the integration of a pre-existing simulation framework was still desirable. It was expected that, aside from making the system more robust and scalable, employing an established framework used and developed by simulation experts would address the management of I/O. It was also hoped that it would benefit in the process of conceptual and simulation modelling, considering that even a generic framework would impose certain rules and standards to which a simulation implementation would have to adhere.

4.2.1.1. Search for an existing implementation

Thorough Internet searches encountered several open source (or otherwise free) frameworks developed in Java, many of which were inclusive of good documentation and demonstration applications. A range of these (including all

frameworks evaluated by Tobias and Hofmann, 2004) were downloaded and inspected further; either by ‘face value’ (e.g. demos and documentation) or more closely by integrating the frameworks into small projects so as to assess their capabilities and suitability for use in the VCS.

Because early efforts failed to identify any appropriate platforms that met the needs of the VCS, the researcher decided to seek expert advice from the field. An email was sent to a Professor at a Technology Institute in Sweden, whose publications of some insightful articles on social simulations suggested he would be a good source for advice. The professor, who kindly responded to the email, was of the opinion that the project seemed to be very complex. He did not know of any tool or platform suitable for the project, and he cautioned that developing a GUI for the simulation could get very complex as well (see Appendix C for the email correspondence).

An unpublished paper on a classroom simulation by an Associate Professor from a University in Florida (USA), suggested valuable input as to how to implement the simulation framework. Since no other information about the project could be found in journal databases or otherwise, an email enquiry was sent directly to this person. His response, however, explained that his classroom simulator was a term project for an undergraduate software engineering class. No publications had transpired from the project and the professor did not share any further details beyond this point (see Appendix D for the email correspondence).

Throughout the first year of the candidature, as further searches and prototypes failed to deliver a suitable framework, feedback from the Swedish Professor became more and more substantiated. Typically one or more aspects of the simulation, such as interactivity, custom GUI, real-time factors and continuous and discrete event requirements, were not fully supported by any single product. When a particular framework looked promising, the researcher would attempt to contact the developer(s) for more information, as was the case with J-Sim (developed at the University of West Bohemia, Pilsen, Czech Republic — not to be confused with a few other simulation frameworks with the same name). Not only did email correspondence with the main developer, Jaroslav Kačer, provide useful information about the J-Sim architecture, but also expert advice about its suitability for the VCS (see Appendix E for the email correspondence). Kačer’s

take on the requirements for the VCS mirrored the opinions of the Swedish Professor; complex and not suitable for the purposes specified. Based on feedback from the domain experts, the decision was made to develop the framework from ground up.

Up to this point in the discussion, a few references to the simulation terms ‘continuous’ and ‘discrete event’ have been made, though no further elaboration of their significance has been given. Fahrland (1970, p. 62) provides a definition of the terms that is quite apt for this research:

The important difference between continuous and discrete event simulations is that in the former all changes may occur at all times for all states of the system, and in the latter changes called discrete events occur at some instants of time for some of the states of the system.

An implicit requirement of the VCS is the allowance of changes to occur continuously (e.g. distractibility is re-calculated on a per-second basis) as well as discretely (e.g. teacher tells student to pay attention whilst in IDLE state). The VCS is a human-in-the-loop simulation where discrete events are dictated by user input, whilst continuous system change occurs automatically at a regular time interval. Thus, these requirements had to be accommodated in the development of the simulation framework.

4.2.1.2. Development of a new simulation framework

Although no suitable ready-made framework was discovered, the research, evaluation and assessment of a number of simulation architectures did improve the researcher’s conceptual and practical understanding in this area. Further, development of the simulation models, as well as GUI design features, had by this stage enjoyed significant progress. Thus, when re-assessing the way in which the VCS was expected to function from a modular perspective, the requirements of the projected framework did not appear to be too overwhelming. Essentially, classroom interaction occurs over time and the periods of interaction are important. Clock and event managers are essential components as they enable the capture of activities and states throughout the simulated session. Hence, the primary modules identified for the framework were:

A clock running in actual (wall clock) real time that can be started, paused and stopped.

An event manager to manage real time information about the simulation, such as status (e.g. running, paused), event states (e.g. IDLE, QUESTION, INTERACTION) and question details (e.g. difficulty, novelty, payload). Most importantly, the manager should provide event subscription and broadcasting facilities (facilitate communication with student entities) and control simulation data storage (I/O).

A simulation manager in control of simulation initialisation (e.g. load and create students, control the clock) and to command the simulation status (e.g. running, paused, stopped).

There are, of course, a number of complexities implicitly related to all of the above modules, but as a starting point this provided a platform upon which development could begin and take shape. To implement the basic requirements of a clock, for example, is relatively straightforward and can with little effort be extended to provide additional services for the managers. A number of sketches and conceptual maps were created and continuously refined throughout the development process. Some were used to plan and describe specific segments of the system (e.g. I/O), while others provided broader overviews of connected components and their interaction (e.g. flowcharts from simulation start to end). Provided in Figure 4.1 is a flowchart of activities, sequences and state transitions implemented in the current system. The figure also indicates the data storage operations that occur in the VCS throughout the simulation run.

The sections below detail what are considered significant modules/aspects of the simulation framework, how these were implemented and the rationale behind the particular design.

4.2.2. Data Input/Output and Data Management

One of the major challenges in the software development process was to identify and implement a system for managing simulation data. With countless methods and technologies for structuring and performing I/O operations, the task of finding

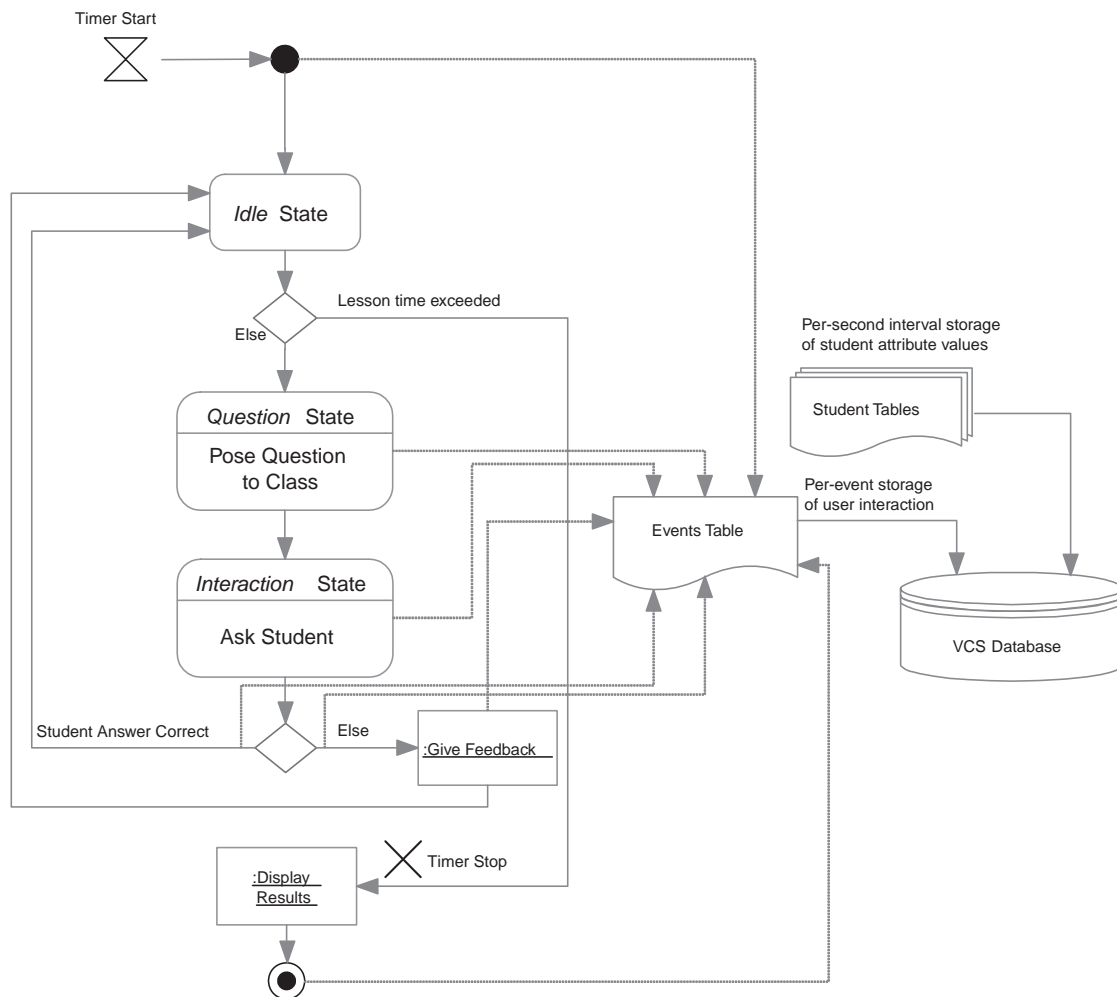


Figure 4.1.: Simulation Flowchart

a “best-fit” solution was reworked a number of times. Simulation I/O extends further than simple procedures for storing, retrieving and managing data, a point well illustrated by Zeigler et al. (2000); it highlights the distinction between internal system structure and external behaviour, where the latter refers to the relationship between the time histories of input and output (see Fig. 4.2).

In other words, system communication and processing need to take into account the significance time and nature of input have on the resulting output. In the VCS, for example, it is crucial that data output maintain complete record entries consisting of time, input parameters and resulting behaviour/state.

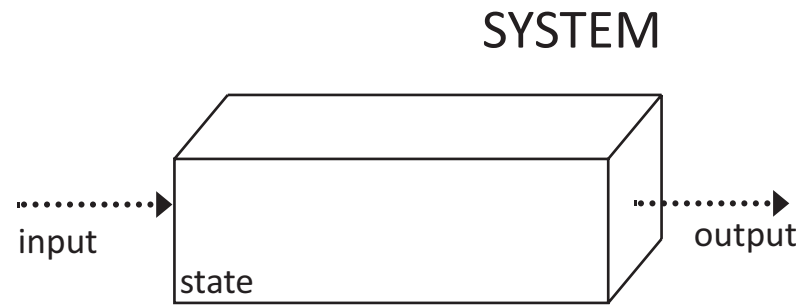


Figure 4.2.: Basic system concepts (adapted from Zeigler et al., 2000, p. 4)

The internal system structure, which concerns the state and state-transition mechanism as well as the state-to-output mapping (Zeigler et al., 2000), will be explained in more detail later in this chapter. The focus of this section is concerned with I/O data management and how this was achieved.

Java DB (Apache Derby) Although, as previously discussed, the XML format was not considered well-suited for VCS data management, it was still standing as the better option from previous work. As such, efforts were made to find ways to structure internal data segments as well as logically split information into files and directories in a modular fashion. The different approaches were also partially tested in software to determine their practicality (e.g. buffering data output vs. continuously writing to file, multiple file access, relationship of data types, importing and structuring results, etc.). Although a well-designed configuration was developed and partially implemented, the amount and types of data expected to be produced and managed reiterated that XML was not an ideal solution. Since the framework had not yet been developed, the data management system could not be stress-tested with real data sizes and at real read/write frequencies. It was nonetheless clear that the approach, even if working, would be cumbersome and bug/error prone.

A great relief came with the launch of Java 6 in December 2006, which included Java DB, a portable implementation of the open source DBMS Apache Derby. Apache Derby, which prior to 2004 existed as a proprietary product called IBM Cloudscape, offers a fast and standards-based database management system with a very small footprint (~2MB) that runs in the same JVM (Java Virtual Machine) as the software in which it is implemented. The system is embedded by adding the

Java ARchive (JAR) file to the project's classpath and its inclusion is transparent to the end user (e.g. no extra installation or configuration required).

Work on implementing data management with Java DB started immediately, with remarkable progress. The system delivered on all promises with very little complexity in the programming of code. The data structures had already been worked out conceptually, so it was only a matter of transferring this knowledge into a database schema (tables, fields and relationships) in the DBMS (see Figure 4.3). Storing or retrieving data was now as simple as running SQL statements (e.g. Algorithm 4.2 on the facing page), with the Java DB automatically taking care of file I/O operations.

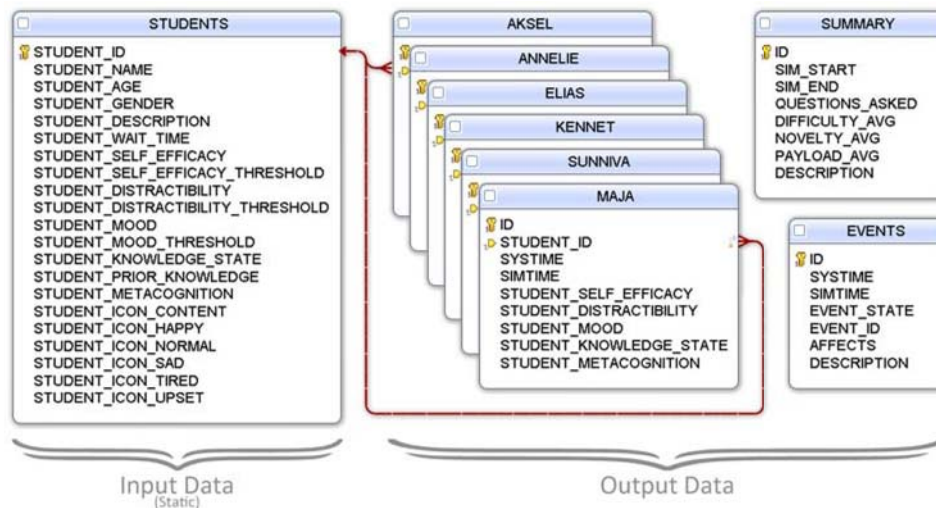


Figure 4.3.: VCS Database Schema

4.2.3. Framework Components

Several Java classes were needed to operationalise the framework and the simulation as a whole. The following is a rundown of the essential modules developed for this purpose. The design and implementation of the GUI is discussed in Chapter 5.

4.2.3.1. SimDBGlobals

The Simulation DataBase Globals (`SimDBGlobals.java`) class contains the SQL statements for generating and populating the database tables used by the simulation (e.g. Algorithm 4.2). Specifically, the included statements create the `EVENTS`, `STUDENT` (one table for each student), `SUMMARY` and `STUDENTS` tables (see Figure 4.3). Only the `STUDENTS` table is populated at start-up, with one record per student containing static, student-specific records used as initial values. Variable student attributes are later stored in their individual tables throughout the simulation run. The `EVENTS` table contains fields for storing event-specific information as they occur in the simulation, whilst the `SUMMARY` table hosts a log of descriptive data (populated at the end of the simulation).

Algorithm 4.2 SQL Statement for the Events Table

```

1 /** Java formatted SQL of Event table STRUCTURE */
2 public static final String EVENT_TABLE_SQL =
3
4 "CREATE table APP.EVENTS ( \n" +
5 " ID                INTEGER NOT NULL \n" +
6 "                  PRIMARY KEY GENERATED ALWAYS AS IDENTITY \n" +
7 "                  (START WITH 1, INCREMENT BY 1), \n" +
8 " SYSTIME           TIMESTAMP NOT NULL, \n" +
9 " SIMTIME           TIME NOT NULL, \n" +
10 " EVENT_STATE       INTEGER NOT NULL, \n" +
11 " EVENT_ID          INTEGER NOT NULL, \n" +
12 " AFFECTS           INTEGER NOT NULL, \n" +
13 " DESCRIPTION       VARCHAR(500) NOT NULL );";

```

4.2.3.2. SimClock

The Simulation Clock (`SimClock.java`) is a timekeeping module which runs in a separate thread and provides essential simulation event controls such as ‘Start’, ‘Pause’ and ‘Stop’ (accessible and controllable by the `SimManager`). Other services include the global provision of elapsed simulation time and facilities for dialogs that are controlled by time.

Every second, the clock will consult the `EventManager` for the current simulation state (e.g. `IDLE`, `QUESTION` or `INTERACTION`) and subsequently

request that this state is broadcast by the EventManager to its listeners. Different processes are executed depending on the current simulation state (e.g. if IDLE, student distractibility is regularly calculated updated). Another important task of the clock is the management of the simulation ‘sweeps’. If the simulation is running, a ‘student sweep’ request is passed on for the EventManager to broadcast to its subscribers. The sweep is performed for the purpose of updating the student database tables with the latest student attribute values. Since this occurs on a per-second basis, a 5-minute lesson will result in 300 record entries per student.

4.2.3.3. EventManager

As the name suggests, the Event Manager (`EventManager.java`) is responsible for the management of event-related properties and processes in the framework. This involves a number of tasks that are pivotal for the simulation framework.

Perhaps the most important task of the EventManager is that of event broadcasting. It defines a map of numerical codes that are used for setting, getting and broadcasting system states and events. Each code in the map, as shown in Algorithm 4.3 (which contains all possible events that may occur in the system), is coupled with a signifier which eases the reading of simulation output (e.g. simulation logs).

Algorithm 4.3 Mapping of Event Codes and Descriptors

```
SIM_SYS_EVENT_STOPPED, "Simulation Stopped"
SIM_SYS_EVENT_STARTED, "Simulation Started"
SIM_SYS_EVENT_PAUSED, "Simulation Paused"
SIM_SYS_EVENT_RESUMED, "Simulation Resumed"
SIM_STATE_IDLE, "Simulation State: Idle"
SIM_STATE_QUESTION, "Simulation State: Question"
SIM_STATE_INTERACTION, "Simulation State: Interaction"
SIM_STATE_CHANGED, "Simulation State Changed"
SIM_EVENT_ASK_QUESTION, "Teacher asked a question"
SIM_EVENT_ASK_STUDENT, "Teacher asked student a question"
SIM_EVENT_GIVE_ANSWER, "Teacher gave answer to question"
SIM_EVENT_GIVE_FEEDBACK, "Teacher gave student feedback"
SIM_EVENT_STUDENT_HAND_RAISED, "A student raised his/her hand"
SIM_EVENT_STUDENT_ANSWERED, "A student answered a question"
SIM_EVENT_GET_ATTENTION, "A student was asked to pay attention"
SIM_STUDENT_SWEEP, "Student Sweep"
```

4.2. Current Development

Components in the framework can subscribe to one or more of these events (codes), allowing for instant alerts in the case of a new event occurring. For example, every Student instance would subscribe to be alerted by every SIM_EVENT_ASK_QUESTION event (a unique final variable of type int), which indicates that a new question has been asked to the class by the teacher. This particular occurrence originates from user input in the Task Panel's GUI, as shown in Figure 4.4.

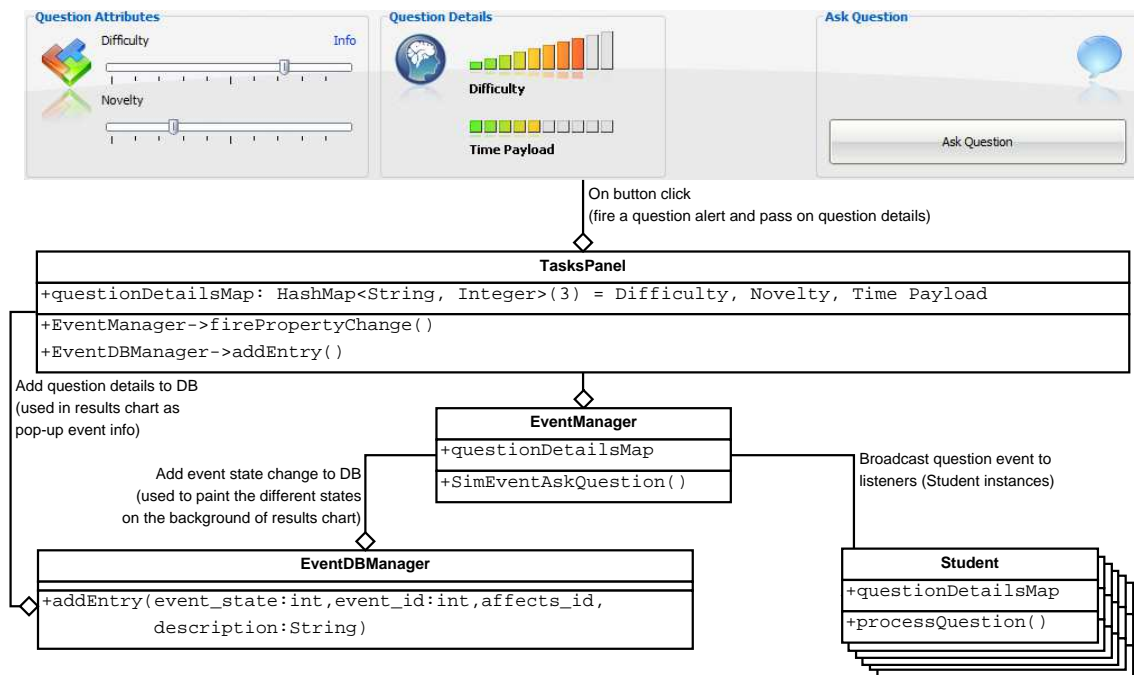


Figure 4.4.: Overview of the Broadcast and Logging of a Question Event

It is the Tasks Panel's responsibility to signal to the EventManager that the event has taken place, e.g.:

```

EventManager.getInstance().firePropertyChange(this,
EventManager.SIM_EVENT_ASK_QUESTION, questionDetailsMap)
  
```

As can be seen above, the question details (e.g. the **questionDetailsMap**, which contains the question's difficulty, novelty and calculated time payload) are

passed on to the EventManager along with the identifying event code. When the EventManager receives the request from the Tasks Panel, it will promptly broadcast the event and question details to all listeners/subscribers (in this case all Student instances). Subscribers will process the information according to the routines defined for handling the event (e.g. process question).

There are also some important logging processes that need to occur in the above example. When a question is asked, the Tasks Panel will communicate directly with the Event Database Manager (EventManager) to request that the event type, time and question details be recorded in the Events table. This information is later extracted and added to the results charts, allowing the user to click on individual event ‘hot-spots’ that display details about questions asked at any particular time. The EventManager will also request that the EventDBManager records an entry in the Events table. This entry request is different in nature to that of the Tasks Panel in that it is specific to the change of simulation state (`SIM_STATE_CHANGED`). Entries of simulation state changes are also utilised in the results charts, where the information is illustrated as different segments on the chart background (as discussed in Section 3.4.5, Chapter 3).

4.2.3.4. EventDBManager

Programmatically speaking, the Event DataBase Manager, or EventDBManager (`EventDBManager.java`), is a very simple class to implement. Its responsibilities include connecting to the database, creating the Events table (executing the SQL statement in `SimDBGlobals`, Algorithm 4.2) and providing a method for adding entries to the table. The EventDBManager is instantiated and controlled by the EventManager who provides a public getter method for instances that need to access event-related logging features.

4.2.3.5. SimManager

The Simulation Manager (`SimManager.java`) instantiates and provides access to a number of the SimClock services (e.g. user input for starting, pausing and stopping the simulation must go via the SimManager). It provides globally available information about the status of the simulation (e.g. if it is running,

paused or stopped) and updates GUI components accordingly (e.g. status display, enable/disable components, etc.).

Instantiation and access to services provided by the `SimDBManager`, a utility class for database connection and interaction, is also handled by the `SimManager`. Final simulation summary is output to a separate table by the `SimManager` via this class.

4.2.3.6. `SimDBManager`

The majority of database-related activities are handled by the Simulation DataBase Manager (`SimDBManager.java`). Its responsibilities include establishing the initial connection to the Java DB driver and to subsequently create and store database and tables to the file system. Since a running simulation commands continuous database access, these tasks must be completed prior to the commencement of the lesson. The `SimDBManager` defines a unique storage path for each simulation using a combination of the home folder path (as determined by the host operating system, e.g. `C:\Users\simon\`), sub-folders dedicated to VCS data (i.e. `vcs_data\vcs_saved_sims\`) and a constant signifier for a VCS database (`vcsdb_`) with a postfix of current system time and date (i.e. `dd_MM_yyyy_HH_mm_ss`). Thus, on a Windows Vista operating system, a path to a folder containing data for one VCS simulation run would look like this: `C:\Users\simon\vcs_data\vcs_saved_sims\vcsdb_18_02_2009_14_20_15`.

While read and write operations to and from database tables are very quick, the one-off initialisation processes are quite time consuming. On a relatively recent system with a dual-core processor (2 GHz) and 2GB of RAM running Windows Vista and Java Runtime Environment 6, these processes take approximately 15 seconds to complete. In earlier prototypes, database initialisation was performed at the start of the VCS lesson (i.e. when the user clicks the start button). This is not an ideal solution, however, as it results in a prolonged wait at a time when the user should get started with the lesson instantly. A key design decision of the VCS was therefore to handle database driver connection and database/table creation at the point of loading the software instead. This method, however, introduces a couple of issues in the event that the user does not actually run a simulated lesson. Aside from the

Chapter 4. Simulation Framework

imposed delay at application start-up, the ~90 database-related files (totalling ~2MB of storage usage) created in the initialisation process would effectively be rendered obsolete. This issue is addressed in the current version by adding measures to ensure that all database files are deleted from the system in the event that the simulation is not run.

In order to keep the user informed about the progress of loading the software, the SimDBManager displays a graphic on the screen (i.e. a ‘Splash Screen’) before the main application window appears. Overlaid on the graphic is a continuously updated text field that informs the user of the progress of loading the application. More details about the splash screen and its implementation can be found in Section 5.3.1, Chapter 5. Figure 4.5 provides an overview of some of the objects and methods involved for starting the application, displaying the splash screen and initialising the database.

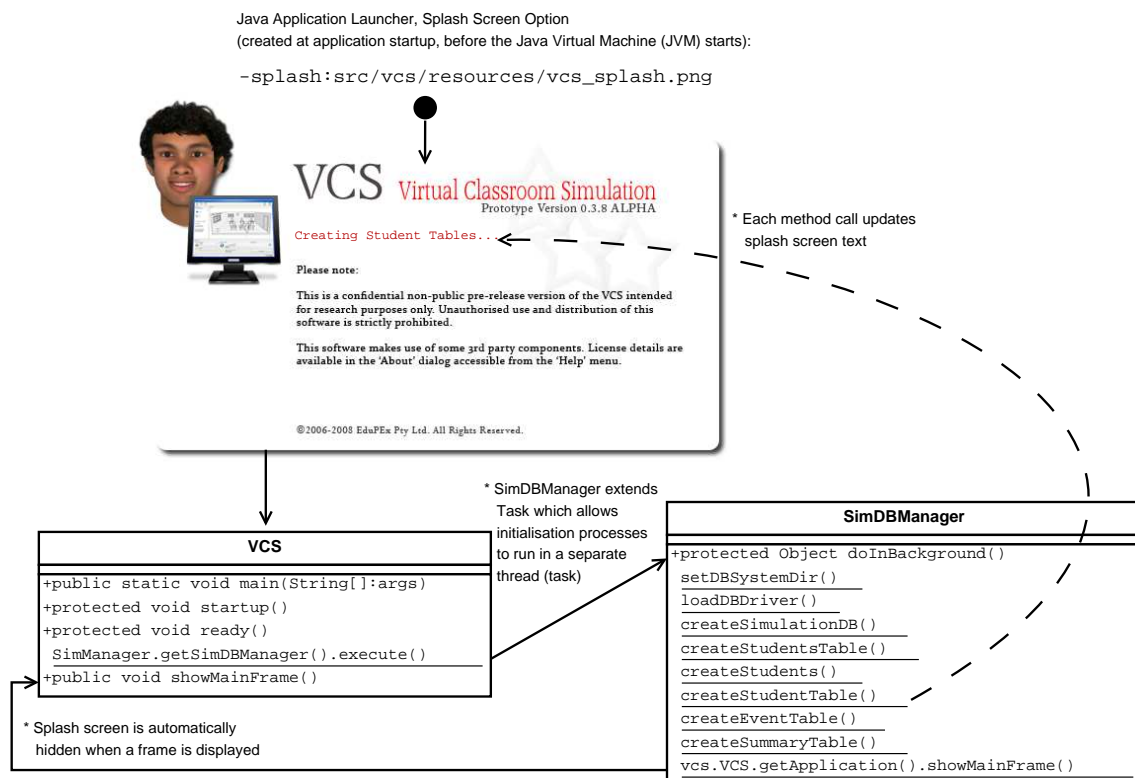


Figure 4.5.: VCS Database Initialisation Overview

Once the SimDBManager has completed its database initialisation tasks it calls

back to the main class with a request to display the initial window of the application. At this point, the splash screen is automatically disposed of and the VCS is ready for user interaction.

4.2.3.7. TasksPanel

Implemented in the current version of the Tasks Panel (`TasksPanel.java`) are both the GUI and the logic for question (task) composition. A question is comprised of three values; difficulty, novelty and time payload. The first two values are manipulated by the user via the slider components located on the panel. The time payload, which signifies how many seconds it will take to ask a specific question, is calculated when the user clicks on the 'Ask Question' button. The algorithm couples novelty and difficulty values with a random factor in order to produce the time payload; thus two questions with the same difficulty and novelty settings may therefore produce a different time payload. Section 4.2.3.3 described the Tasks Panel's involvement in the firing of a question event, change of simulation state and logging processes. The conceptual model on which the panel is based can be found in Section 3.4.3.2 (Chapter 3) while the GUI implementation is described in Section 5.3.4.3 (Chapter 5).

4.2.3.8. Student

Each Student (`Student.java`) instance is created by the `SimDBManager` on application start-up (Section 4.2.3.6), subsequent to it creating and populating the students table defined in `SimDBGlobals` (Section 4.2.3.1). As soon as this table has been populated the `SimDBManager` immediately connects to it and loops through the rows, each of which contains the information needed for instantiating one Student. Algorithm 4.4 shows the section in the loop where data retrieved from the Students table (`java.sql.ResultSet rs`) is passed to the Student class' constructor.

The resulting instance is stored in an array (`Student students[]`) of one or more Student instances. The size of the array, hence the number of students in the class, is dictated by the number of rows retrieved from the `STUDENTS` table. Student values are not retained between lessons, meaning that the starting point of

Algorithm 4.4 Student Instantiation

```
students[i] = new Student(  
rs.getInt("STUDENT_ID"),  
rs.getString("STUDENT_NAME"),  
rs.getInt("STUDENT_AGE"),  
rs.getInt("STUDENT_GENDER"),  
rs.getString("STUDENT_DESCRIPTION"),  
rs.getDouble("STUDENT_WAIT_TIME"),  
rs.getDouble("STUDENT_SELF_EFFICACY"),  
rs.getDouble("STUDENT_SELF_EFFICACY_THRESHOLD"),  
rs.getDouble("STUDENT_DISTRACTIBILITY"),  
rs.getDouble("STUDENT_DISTRACTIBILITY_THRESHOLD"),  
rs.getInt("STUDENT_MOOD"),  
rs.getDouble("STUDENT_MOOD_THRESHOLD"),  
rs.getDouble("STUDENT_KNOWLEDGE_STATE"),  
rs.getDouble("STUDENT_PRIOR_KNOWLEDGE"),  
rs.getDouble("STUDENT_METACOGNITION"),  
rs.getString("STUDENT_ICON_CONTENT"),  
rs.getString("STUDENT_ICON_HAPPY"),  
rs.getString("STUDENT_ICON_NORMAL"),  
rs.getString("STUDENT_ICON_SAD"),  
rs.getString("STUDENT_ICON_TIRED"),  
rs.getString("STUDENT_ICON_UPSET"));
```

every simulation session is identical. For future versions it has been considered to implement a ‘student editor’ which allows the user to manually manipulate student attributes prior to commencing the lesson. It has also been considered to implement a feature which carries over student attributes from one session to the next.

Once a Student is created, the SimDBManager also instantiates a Student DataBase Manager (`StudentDBManager.java`) for that particular Student. In turn, the StudentDBManager creates an empty table for its associated Student in which attribute changes will be stored throughout the simulation. Figure 4.3 (on page 82) showed the six tables created for student output and the fields defined within these.

Individual Student instances are responsible for creating and updating their own Student Panel (`StudentPanel.java`). This GUI component (discussed further in Section 5.3.4.1, Chapter 5) is in need of continuous updates to reflect mood (student icon), distractibility (vertical coloured bar), state (thinking, hand raised, answering) and the enabling/disabling of user input (get attention and ask question).

A Student instance subscribes to a number of events defined by the EventManager and is thus able to make appropriate visual updates to the Student Panel whenever a new event is broadcast. The events subscribed to by the Student are outlined below:

SIM_STUDENT_SWEEP This event is fired once per second whilst the simulation is running. It prompts each Student instance to add a new entry to its database table with its current attribute values (self efficacy, distractibility, mood, knowledge state and metacognition).

SIM_STATE_CHANGED If the newly entered state is IDLE, logic is put in place to start processing the student's distractibility. If the newly entered state is not QUESTION, visual indicators in the Student Panel are updated.

SIM_STATE_IDLE The IDLE state is broadcast every second. Whilst in this state, the listener will make a call to a method named `processIdleState`, which updates the student's distractibility on a per-second basis. Students become distracted at a different rate, but if the teacher leaves the class idle over a prolonged period of time students will become too distracted to process questions. The 'Get Attention!' action in the GUI will alleviate the distractibility levels. The IDLE state is the only state in which the lesson may terminate. Thus, if the predetermined lesson duration (e.g. 5 minutes) is reached whilst in a different state, the lesson will continue until the IDLE state is again reactivated. This is to ensure that any pending question/answer/feedback processes may be completed before the simulation ends.

SIM_EVENT_ASK_QUESTION This event is fired once when a question is asked to the class. Upon receiving this event, each Student instance will collect the question details attached with the event. These are used in the processing of the question, which will determine whether or not the student is capable of answering the question and the wait time. In turn, the process will also update visual components of the Student Panel and make appropriate entries to the database. Figure 4.6 shows an overview of the processes and decisions involved when a question is asked to the class.

Figure 4.7 continues to illustrate what happens when a student is asked to answer the question.

SIM_STATE_QUESTION This state is entered when the `SIM_EVENT_ASK_QUESTION` event is broadcast. The `QUESTION` state fires on a per-second basis until the teacher asks a student to answer the question. Processing occurs when the teacher is no longer asking the question, upon which a countdown of wait-time is activated (if the student was able to process the question). When the wait-time has counted down to zero, visual updates are made on the Student Panel to signify that the student is ready to answer the question.

SIM_EVENT_ASK_STUDENT This event is broadcast when the teacher selects a student to answer the question. A set of processes will be activated for this particular student in determining the final outcome (including a number of database entries). The performance of the student (and potentially teacher feedback) will also affect all other students in the class, as shown in Figure 4.7. The figure, which uses student Maja as an example, outlines the flow of decisions and processes that occur in the event of a student being asked to answer a question. Maja would in this example be assessed on a number of criteria before an answer is potentially given. Should she be too distracted or the question too difficult for her knowledge state, it may have a negative effect on her mood and self efficacy (after all, she did not indicate that she was willing to answer the question). A different scenario occurs if Maja is paying attention and her knowledge state is on par with the question difficulty, but the teacher asks her to answer before she is ready. In this case her mood may be affected negatively, however she may still be able to answer correctly (depending on a number of criteria). Final scenario is when Maja's hand is raised and she is asked to answer the question. If she answers correctly, her mood and self efficacy will increase. Other students (who are paying attention) may either experience an increase in their self efficacy and mood at varying levels (if they also knew the answer) or knowledge state (if they did not know the answer, but learned from Maja's correct response). In the event that Maja did not

SIM_EVENT_ASK_QUESTION

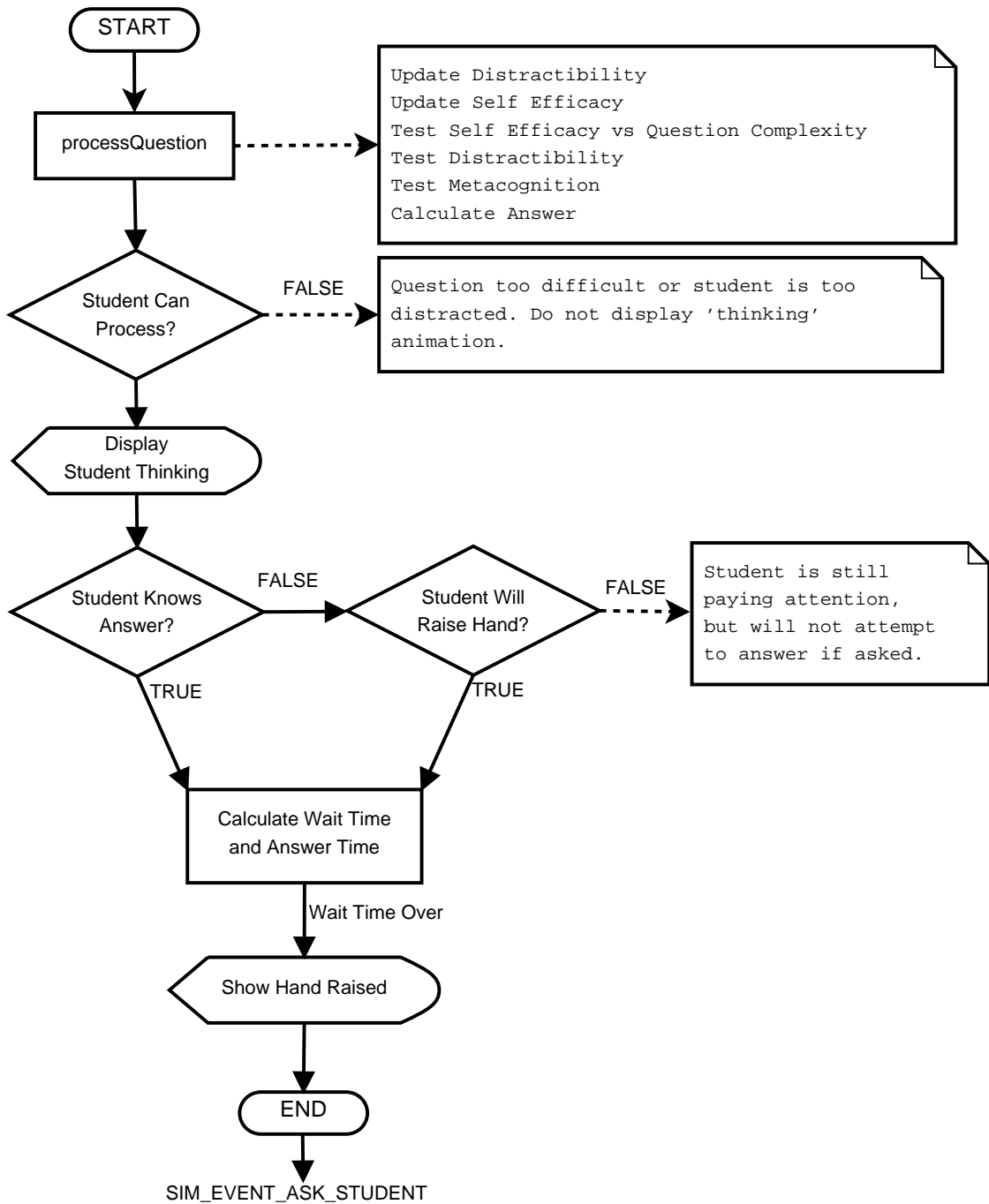


Figure 4.6.: Overview of the Ask Question Event

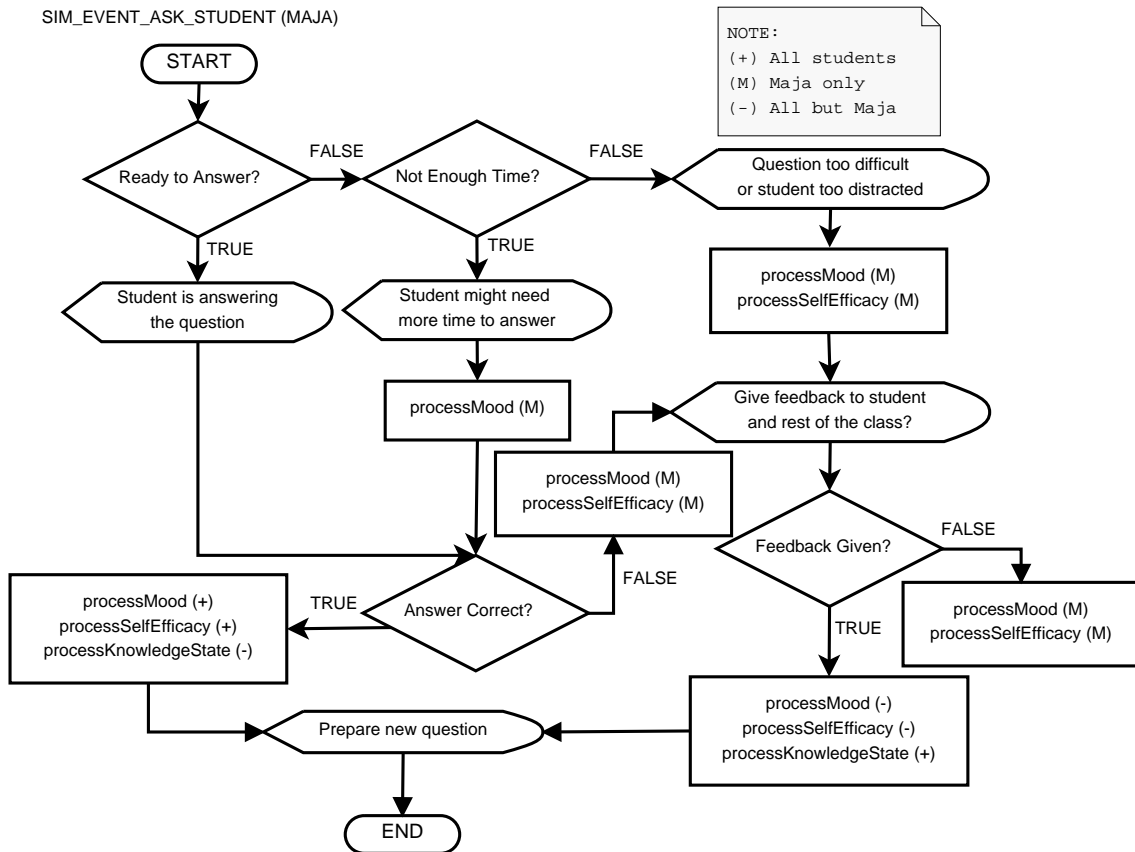


Figure 4.7.: Overview of the Ask Student Event

know the answer, the teacher is presented with an option to provide feedback (at a time penalty). Should the teacher choose not to give feedback, Maja may experience a further blow to her self efficacy and mood. If feedback is given, on the other hand, similar processes to that of Maja answering correctly will occur although Maja's mood and self efficacy will not be affected further. Knowledge state for all students, including Maja, may also increase.

4.2.3.9. StudentDBManager

Each Student has associated with it an instance of the Student DataBase Manager (`StudentDBManager.java`). Aside from providing a connection to the simulation database (facilitated through the `SimDBManager`), the `StudentDBManager` creates and accommodates for write access to a Student table (each Student has its own table). This service is heavily used by the Students for logging their attribute values throughout the simulation. The resulting dataset is subsequently used to draw the graphs in the results chart.

4.3. Other Software Features

The VCS software includes a number of features and services that are not directly involved with the core components of the simulation framework. They do, nonetheless, interact with the framework and/or otherwise add value to the software implementation. The following subsections acknowledge the contributions from some of the more significant components in this regard.

4.3.1. VCS Console

The VCS Console (Figure 4.8) provides a detailed account of events and entity data as they occur in the system.

The information output to the console is presented in plain text and contains more details than what is stored in the database. While the information is not useful to the end-user per se, it provides the developer with important data used for tracing and validating computational output. Errors (exceptions) caught by

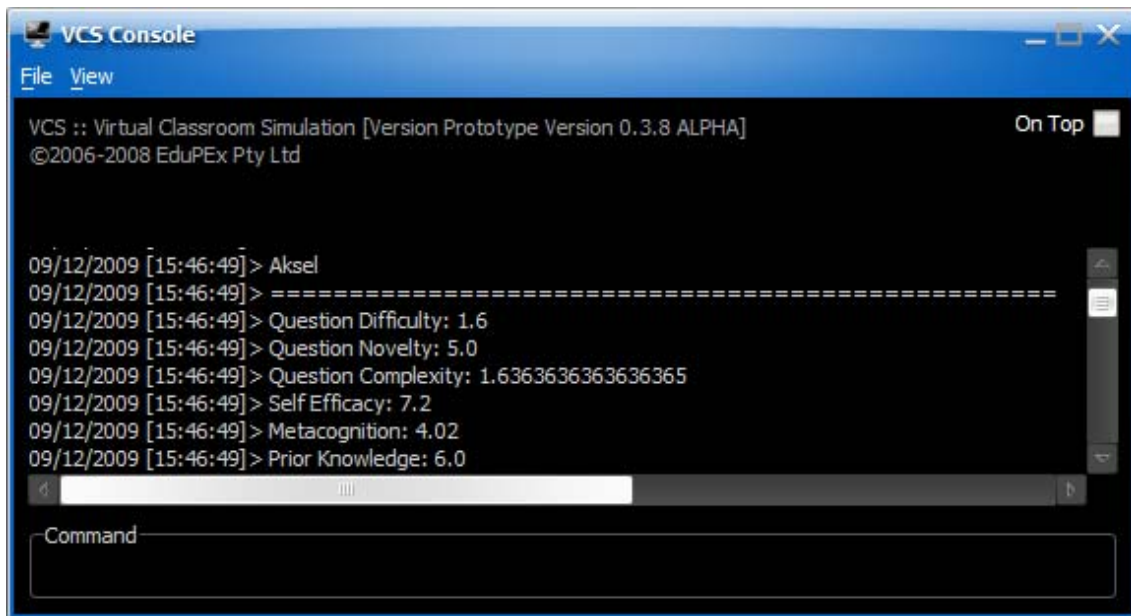


Figure 4.8.: VCS Console

custom exception handlers defined in the source code will also be output to the VCS Console, thus adding further value to the development process.

4.3.2. Persistence

Simulation-related data are automatically stored ‘on-the-fly’ (e.g. spontaneously) to a database unique to that session. However, if the VCS is terminated before the lesson has completed (e.g. in the event of a computer crash or manual exit), that particular lesson cannot be resumed at a later stage. Further development in this area is planned for future work.

A results viewer (‘VCSChart’), similar to the results view in the VCS, has partly been implemented as a separate application that enables the user to view charts from past simulation runs. The viewer is also able to read and report on data from incomplete lessons. Development of the viewer was not essential for this research project and put on hold due to time-limiting factors. Completing the application is planned for future work.

Persistent GUI session state is automatically handled by the Swing Application Framework, as described in the next section.

4.3.3. Swing Application Framework (JSR-296)

The VCS makes use of the Swing Application Framework (part of the Java Specification Request (JSR) 296), a prototype implementation that simplifies building desktop applications by providing an infrastructure that is common to most desktop applications (Potochkin and Muller, 2007), e.g.:

- Application lifecycle, notably GUI start-up and shutdown.
- Support for managing and loading resources, like strings, formatted messages, images, colours, fonts, and other types common to desktop applications.
- Support for defining, managing, and binding Actions, including Actions that run asynchronously (in the "background").
- Persistent session state: support for automatically and selectively saving GUI state from one run of an application to the next.

4.3.4. User Confirmation

A number of measures have been put in place to ensure that the user does not inadvertently perform actions that may affect the operation of the system. Confirmation dialogs have therefore been implemented for critical actions, such as exit the program or stop the lesson (see Figure 4.9). Similarly, if the user opens the results window, the simulation will automatically pause and user input to the main window is blocked.

4.3.5. Modern Heap View

The VCS employs Scott Violet's source code for a Modern Heap View (MHV). The MHV module, which is located in the status bar of the main window, is a continuously updated visual indicator of memory usage and allocation (see Figure 4.10).

Additionally, a mouse click on the MVH forces the Java garbage collector (`java.lang.System.gc()`) to be executed. This induces the Java Virtual Machine "to recycle unused objects in order to make the memory they currently occupy

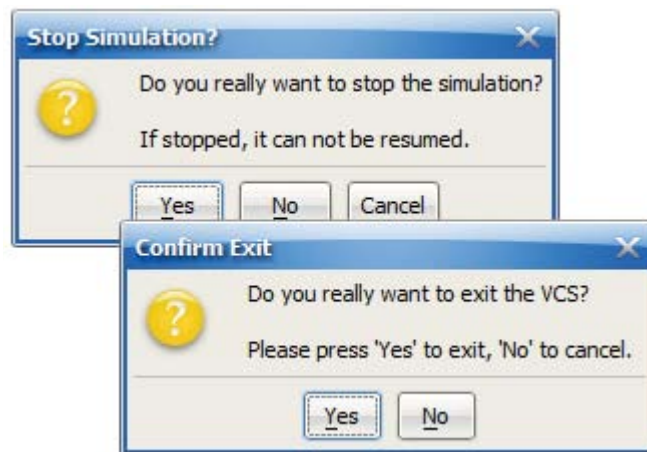


Figure 4.9.: User Confirmation Dialogs

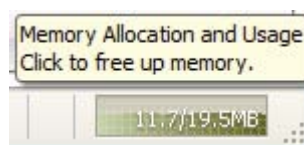


Figure 4.10.: Modern Heap View Component

available for quick reuse. When control returns from the method call, the Java Virtual Machine has made a best effort to reclaim space from all discarded objects” (Sun Microsystems Inc., 2008).

The module was initially intended to be used only for memory monitoring purposes during development of the VCS. It has been kept in place in the current version as it is a useful utility that allows users to monitor memory consumption as well. More information about the MHV can be found in Appendix A, Section 301.

4.4. Third Party Technologies

The School of Education, the University of Adelaide, provided licenses to some commonly used software (e.g. office software, productivity and data analysis tools). More specialised technologies related to software development, however, were not readily available. With a limited budget, it was decided that additional resources (including software, modules, packages, code, graphics, etc.) used for

developing the VCS should be open source or otherwise free. Appendix A provides a complete list in which third party technologies utilised in this study are acknowledged. The Integrated Development Environment, a computer application for software development, is one piece of technology that merits further elaboration and acknowledgement.

4.4.1. Integrated Development Environment

Arguably the most important tool for developing the VCS was the Integrated Development Environment (IDE). An IDE is a software application which facilitates and simplifies the development of new software by integrating typically needed components (e.g. source code editor, GUI designer, compiler/interpreter, debugger and database access to name a few). Desirable features for a suitable IDE for VCS development included:

- Non-proprietary licensed
- Support for Java development (preferably with full support for the most recent Java version)
- Advanced source code editor, preferably with
 - Inline parsing
 - Refactoring facilities
 - Code highlighting
 - Code completion
- Debugger and preferably a profiling facility (for monitoring components' CPU and memory usage)
- GUI designer with support for the Java Swing API
- Cross-platform compatible (workstations used in development were running Microsoft Windows and Linux)
- Database service able to interface with JavaDB

After some research, the two IDEs that looked most suitable for the project were Eclipse (The Eclipse Foundation) and NetBeans IDE (Sun Microsystems). Both IDEs were well-developed, community driven, not-for profit and open source projects with a large number of users. The final choice fell on NetBeans IDE 5.0, as it provided all the desirable features listed above. A number of new versions of the NetBeans IDE were released throughout the candidature, providing additional features and support for new versions of Java. NetBeans IDE 6.1 was the last version used in the development of the VCS.

4.5. Summary

Prior efforts in advancing the VCS framework did not identify, nor produce, any tangible technologies or solutions suitable for adoption. They were, however, successful in identifying a number of challenges that had to be carefully considered in the current study. Several simulation frameworks were investigated, trialled and prototyped in the early stages of this research study. Due to the specific requirements implied by the VCS' simulation model, however, none of these were suitable for adoption or adaptation. Advice from experts in the field supported the conclusion to develop a simulation framework from ground-up.

The final VCS simulation framework, in which entities and models of interaction established in the previous chapter were advanced, integrates a number of technologies. These technologies were carefully implemented under the guidance of three main objectives; a) fulfill the requirements of the simulation model, b) address the second problem of this research study and c) provide essential services for the graphical user interface. The framework represents a significant achievement as it meets all three objectives in a convincing manner. To substantiate this claim, however, trials and feedback from the target population were necessary. This makes part of the research design, which will be discussed further in Chapters 6 through to 10.

The next chapter will discuss the design and implementation of the graphical user interface. It will also demonstrate how tightly integrated the GUI is with the simulation framework as well as the human user.

Chapter 5.

Graphical User Interface

It seems that UI are often more difficult to engineer than other parts of the system. UI are critical to the success of software products. However, designing UI is difficult and time-consuming.

(REITERER, 2000, P. 225)

Although much work has been devoted to the research, specification and implementation of conceptual models (Chapter 3), as well as the underlying simulation framework (Chapter 4), these efforts would be of little benefit to the end user had there been no interface to facilitate Human-Computer Interaction (HCI) with the VCS. With a large amount of information needed to be conveyed, both in terms of how to use the program as well as how to make sense of real time data output, planning and implementation of the Graphical User Interface (GUI) became more complex than anticipated. Ways in which to make the graphical representation as attractive and intuitive as possible were also in need for careful consideration. Although the simulation's purpose and target user group had been identified (i.e. education students, see Chapter 1, Section 1.4), little was known about the population's experience, knowledge, preference and attitude towards information technology. These were, in fact, user characteristics that the study sought to investigate *after* the VCS trials had taken place. Nonetheless, the diversity of individuals in the population prompted a user interface suitable for both novices and experts. Development of an intuitive design suitable for serving the envisaged purpose of the VCS became a key aspect of the study.

5.1. GUI Objectives

Optimising the simulation's usability and usefulness, yet minimising GUI complexity, was motivated by two main factors; 1) inclusiveness of ICT proficiencies at varying levels and 2) ease of administration and integration of the VCS into a lesson plan. While system compatibility (e.g. hardware and software) and ease of setup are essential prerequisites for using the software, the first impression of the software itself is another important factor. Norman (1988) reminds us that there is no such thing as the average person and that there are challenges involved with developing a single design that suits everyone. An individual's readiness for adopting a new tool, or the tool's threshold for uptake, is generally subject to a few precursors such as the 'radicalness' or 'newness', learning curve, perceived value, attitudes and so forth. Varying preferences and attitudes towards ICTs may also pertain to course administrators (lecturers and tutors). Thus, the importance of an appealing and easy to use interface applies to educators and education students alike. Additionally, course administrators are as much responsible for adding value to the VCS as the VCS is in adding value to the course. In order to get value out of the simulation, an investment of time and resources must be made. In particular, it is essential that they acquire a sound understanding of the theories and objectives behind the VCS in order to fully understand the potential of the tool, recognise ways in which it may be of value and how to integrate it into the lesson plan.

Shneiderman and Hochheiser (2001) use the term 'universal usability' to describe technology designed in such a way that just about anyone can use and get value out of it. While the user interface of the VCS certainly aims to be universally usable, it should be reiterated that universal usefulness is highly subjective and thus limited to the scope of the target user group. Reiterer (2000) argues that the user interface is critical to the success of a software product and that its engineering is often more complex than other parts of the system. Both in terms of usability and usefulness, this added layer of complexity should be given careful attention in order to ensure that the perceived threshold for using and extracting value from VCS is not too high. Considering the context in which the VCS is to be deployed, these issues become increasingly important to the design and development of the user interface.

In this regard, it is useful to consider some important concepts from the technology adoption life cycle and marketing of technological innovations.

5.1.1. Lowering the Uptake Threshold

The technology adoption life cycle model, which was partly developed (and perhaps made most famous) by Rogers (2003), identifies a number of communities in a population, each of which with a different stance when it comes to the adoption of new innovations, ideas and technologies. The model also suggests the rate of adoption over time as a normal distribution, with the communities typically labelled as innovators, early adopters, early majority, late majority and laggards.

In his discussion of the technology adoption life cycle, particularly with regard to high-tech products, Moore (1995) suggested that the threshold for uptake in the early market, which in the model is made up of innovators and early adopters, is a lot lower than in the mainstream market (early and late majority). In other words, a product with a high and promising rate of diffusion in an early market may not enjoy the same fortune when trying to enter the mainstream market. According to these observations, Moore (1995) introduced the notion of *the chasm*; a term used to describe the gap between the early market and the mainstream market (see Fig. 5.1).

Moore (1995) did also re-label the population segments of the model to make it fit better with the high-tech market:

Technology Enthusiasts = *Innovators* Lovers of technology — the first people to get their hands on the absolutely latest innovations. Moore (1995) describes these people as techies; the one person in your family or at work who knows how to set up a wireless network, to configure the new high-definition television and to troubleshoot problems when your iPod won't sync with iTunes. They are the alpha testers, bug-reporters, code contributors and forum enthusiasts; the ones who love to explore and use new stuff. Although technology enthusiasts do not have money to invest in the market, their distinguished taste has a great influence on the visionaries.

Visionaries = *Early Adopters* Described by Moore (1995) as “the true

NOTE:
This figure is included on page 104 of the print copy of
the thesis held in the University of Adelaide Library.

Figure 5.1.: Divide between markets (adapted from Moore, 1995, p. 19)

revolutionaries in business and government”, the visionaries are concerned with being first to market with, and exploit, new promising technologies so as to gain a competitive advantage. Visionaries have the money to invest in both the product and in marketing.

Pragmatists = *Early Majority* Pragmatists are careful when adopting a new technology, and when they do it is always a well-informed decision. They believe in evolution rather than revolution, and a product needs to have a proven track record, references and be made by the market leader in order to adopt it (Moore, 1995).

Conservatives = *Late Majority* These people do not think much can be gained from new technologies. ‘Pessimistic’, ‘price-sensitive’, ‘sceptical’ and ‘demanding’ are some of the words Moore (1995) uses to describe this group. In order to get through to the conservatives, systems need to be highly simplified.

Sceptics = *Laggards* Not only are they sceptics, but they are also critics and

challengers of new technology. The wisest thing to do with this group is to avoid them altogether and focus on the other market segments.

The above holds relevance to the current study as, on a user level, individuals within the target population may be categorised in similar segments. Software usability is greatly impacted by the user interface, so in order to appeal to a larger market of educators (e.g. cross the chasm), the VCS must be able to clearly and efficiently demonstrate its purpose, use and value. The population's 'technology enthusiasts', for example, may be a good indicator of the usability of the software. They may also provide important views on what technical aspects need to be refined and improved. The 'early majority', on the other hand, may be more likely to reflect on the usefulness of the VCS. Feedback from these groups is a valuable resource that may assist in improving the software through further research and development. The importance of visionaries becomes more apparent in the deployment and uptake of the software, or what Moore (1995) describes as 'crossing the chasm'. In this context, visionaries may be thought of as decision makers in academia who recognise the VCS' value-adding potential for teaching and learning. In order to cross this chasm, it would not necessarily be sufficient to demonstrate successful trials within a group of technology enthusiasts alone. Research and trials should prove that the tool is a manageable resource for course administrators and that it does in fact meet its aims and objectives. It would be difficult to promote the VCS to decision-making pragmatists without support from visionary institutions.

Concepts from the technology adoption life cycle provide useful guidelines for software development and design. They are also relevant in the analysis of survey data. Collected information may be used to identify where participants belong in the market segments and thus how to interpret individual responses (feedback from an identified pragmatist would be given different considerations than that from a sceptic). This research study ensured that information pertaining to individuals' attitudes and comfort with computers was collected. It also employed instruments for assessing attitudes towards the technical, GUI and educational aspects of the VCS. While the final analyses of this data do not explicitly categorise individuals into market population segments, they are nonetheless taken into account.

The next sections will discuss the GUI design in this study and how prior efforts

informed the current.

5.2. Earlier Work

Many of the educational theories advanced in earlier work on the VCS were employed in the current study. The software implementation, however, does not integrate any code, modules, data structures or any other components developed in the period prior to the PhD candidature. The rapid evolution of the Open Source (OS) landscape in particular prompted a re-think of how the VCS software could be implemented. The range, quality and generous licensing of many third party components for software integration, for example, allow for greater flexibility in what can be achieved with limited resources. Integrated Development Environments (IDE) and other production software (e.g. for database development) are also flourishing, with new and improved versions released frequently. The Java Application Programming Interface (API) has also matured over several version upgrades, with a number of added features and improvements.

This is not to say that earlier work on the user interface was without meaning or value to the current efforts. Many areas of the current implementation, including design features and framework logic, can be explained by looking at original approaches to solving the problem. This section will outline some of the key areas from which current GUI development has drawn inspiration and knowledge through experience.

5.2.1. First GUI Efforts

User interface development of the VCS has gone through several stages since 2002, and visual representations of concepts and ideas were utilised from the very beginning. Even the earliest GUI prototypes helped refine and mature ideas with regard to what features should be included and how they could ease HCI. Although these representations were effectively no more than empty ‘shells’, rapid prototyping (e.g. ‘the use of prototypes as a way of testing ideas’, Tripp and Bichelmeyer, 1990, p. 35) of the GUI was nonetheless useful to demonstrate ideas, identify challenges and opportunities and increase creative thinking. Balci and

Nance (1987, p. 755) describe the process of rapid prototyping as a quick and inexpensive way to test ideas, as well as a good way to obtain new knowledge about the target system. Even though several prototypes may be discarded, the knowledge of what does and does not work is retained, thus adding value to new and improved versions.

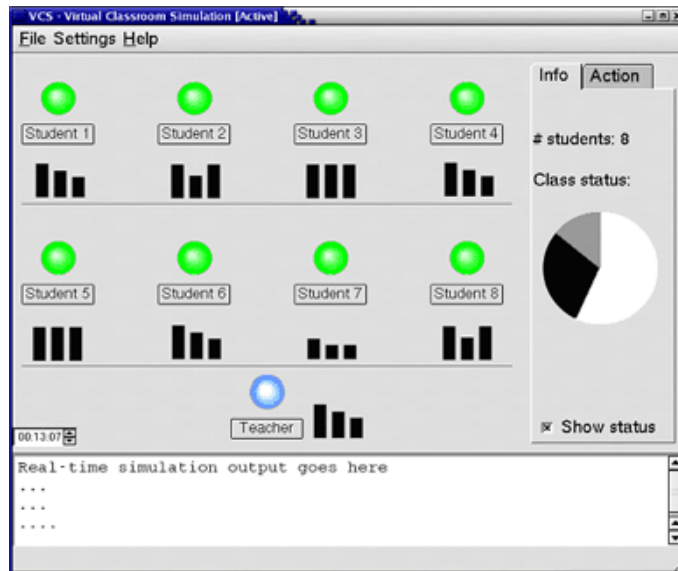


Figure 5.2.: One of the first GUI representations of the VCS classroom (2002)

It was presentations of screens like the one in Figure 5.2 that helped conceptualise many of the features present in subsequent (including the most current) versions of the VCS (e.g. the notion of real-time interaction and the non-deterministic nature of events).

5.2.2. Honours Work

As described in Skrødal (2003, Chapter 5), with more efforts invested in the conceptual modelling and various features and components related to the simulation framework, user interface development was given a lesser priority in the Honours project. It did nonetheless advance ideas and features that influenced decisions in the current research, both in terms of design and functionality.

The final layout of the main window, as shown in Figure 5.3, retained many of

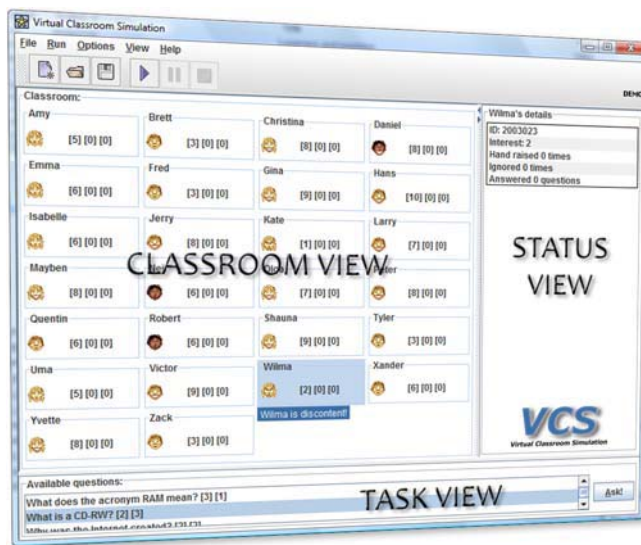


Figure 5.3.: Honours version of the VCS (main window)

the features initially introduced in the first GUI prototypes and would also become an important guide for the current study.

5.2.2.1. Status View

The old status view, as shown in Figure 5.4, contained only basic student information.

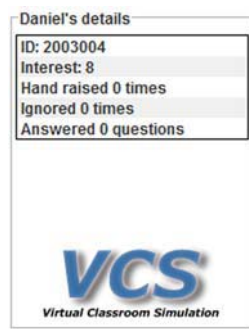


Figure 5.4.: Status view (Honours version,)

A right mouse-click on a student in the class view would populate the data fields with information about that student. Since the simulation framework never reached

a functional state, the information displayed was a static view of what was stored in the student data file (e.g. data would not be updated during the course of the simulation).

It was intended to add more details to this view, particularly with regard to specific information about the simulation run (status, time, etc.). There were also talks about adding more elaborate details about individual student states. As will be discussed in Section 5.3.4.2, this would change significantly in the current study.

5.2.2.2. Task View

The task view displayed a list of questions available to the teacher for the specific subject, as well as a button to communicate a selected question to the class. To give the teacher a cue about the properties of a question, they each included numerical attribute values for the complexity and interestingness (numbers in brackets, as shown in Figure 5.5).



Figure 5.5.: Task view (Honours version)

By providing a number of pre-made questions for a subject in this manner, it would allow the teacher to adapt the lesson to the state of the students/class. For example, if the virtual students were to display a disliking behaviour to the type of questions asked, the teacher could ask questions with alternative attribute loadings.

It was intended that new questions could be created by the users and loaded into the software, allowing virtually any subject to be taught in the VCS. There were, however, a few eminent problems with this approach which will be described in more detail in Section 5.3.4.3.

5.2.2.3. Classroom View

The classroom view, as depicted in Figure 5.6, provided very basic visual stimulus in regard to the students' individuality, mood and so forth.

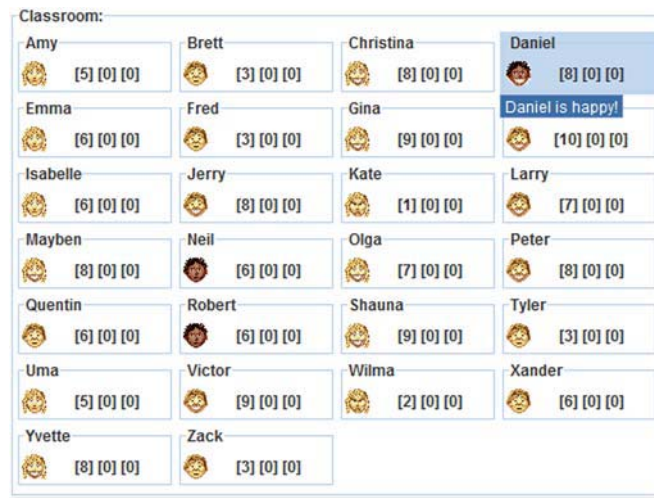


Figure 5.6.: Class view (Honours version)

The graphical representation was disadvantaged both by the researcher's limited skills and the large number of students in the class. Although only three unique student faces (each with three different moods) were provided in the program, the intention was that more be added in future work.

The only direct interaction with individual students was a right-click feature for displaying further information in the status view. This feature added little value since the information replicated only what was already presented in the classroom view. Even though the classroom view had a very basic implementation and representation, some aspects were built upon and improved in the current version.

5.3. Current GUI

Development of the VCS is closely linked and dependent on contributions from a number of disciplines. Thus, difficulties and delays in one discipline will likely affect the progress in others. This was evident in the Honours work, which never produced a functional version. Design and implementation of the GUI, for example, was delayed and thus limited due to issues with the DIAS simulation framework. In the current research study, however, all areas of the software, including the user interface, have enjoyed significant progress. Work on the simulation framework has

inspired and propelled development of the GUI and vice versa. Thus, a number of aspects pertaining to the GUI have changed or otherwise been improved in the current version of the VCS. Some changes were imposed by the functional objectives of the study while others were driven by aesthetics and user-friendliness.

While there are no hard and fast rules on designing the user interface, a *user-centred design* that considers the needs and interests of the user should be followed. Norman suggests that “tasks should be simple in structure, minimizing the amount of planning or problem solving they require” (2009, p. 191) and that the designer should “use constraints so that the user feels as if there is only one possible thing to do — the right thing” (2009, p. 199). This section will outline how Norman’s suggestions were addressed in the GUI design. It will also detail and justify the challenges, features and future direction of the current user interface.

5.3.1. Splash Screen

Many applications include what is typically referred to as a ‘Splash Screen’; a graphic displayed while a software application is loaded into memory (Zukowski, 2006). The splash screen provides a visual indicator that the program is indeed loading and preparing to start. It is an important means for communicating with the user when the resource requirements of the program and/or the performance of the host computer impose a potential delay in start-up time. Some readers may have encountered software applications that do not provide this feature. And some may have attempted to re-load the application one or more times in succession, only to find that when the computer finally responds, a number of instances of the program is loaded.

The VCS requires some time to load the main application as well as create the session’s database and associated tables (creating numerous files and folders on the file system is generally quite slow). Thus, if no splash-screen was provided, the user would typically be left without any indication that the VCS was loading for a period of 10-20 seconds. If, due to this lack of information, the user re-attempt to start the simulation another instance will load, creating another set of database files and folders on the host computer.

In versions prior to Java 6, there were no specific splash screen facilities

provided. Developers therefore had to program their own custom window to display the graphic. Aside from having to code and design the application logic, the custom implementation could not be displayed until *after* the Java Virtual Machine (JVM) was fully initialised. Since the JVM requires some time to load into memory as well, the delay involved in displaying a custom splash screen would be dictated by the host system's available resources.

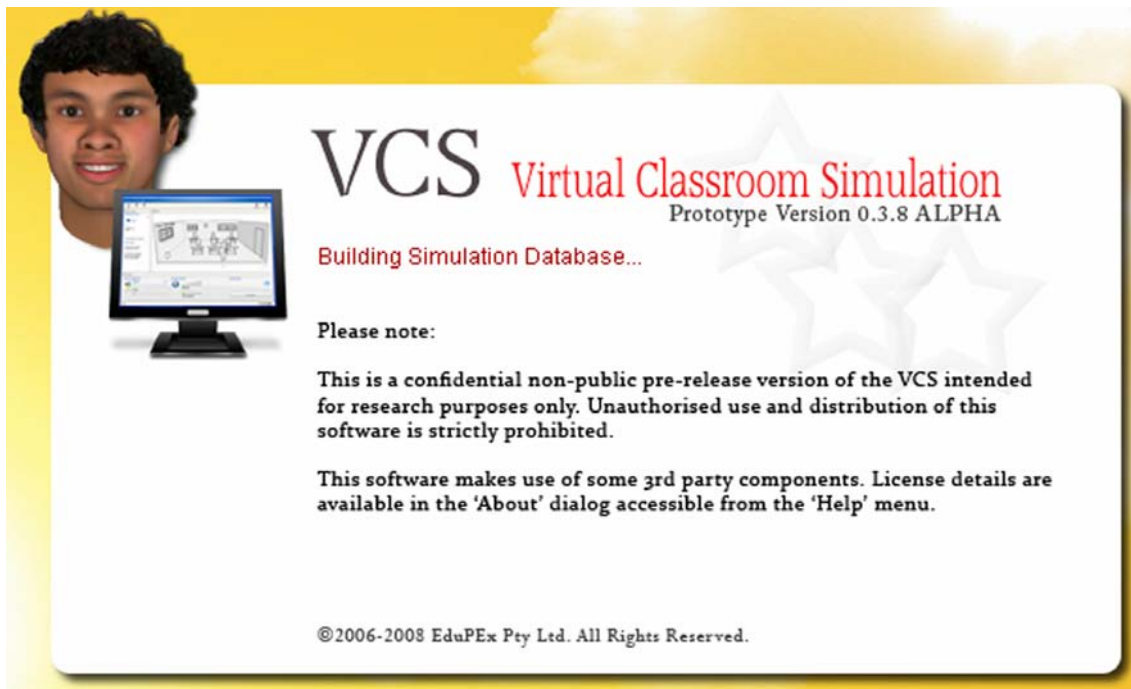


Figure 5.7.: VCS Splash Screen

With the advent of Java 6, the API now includes a new splash screen feature (`java.awt.SplashScreen`) which enables the display of an undecorated window *before* JVM start-up. The new feature does also make several aspects of the actual splash screen implementation easier and more standardised. By the time Java 6 was officially released, development of the VCS was well underway (using Java 5) and a custom splash screen had already been implemented. Since the researcher decided to embrace the new API due to several welcome additions, the old implementation was eventually abandoned and a new developed utilising the splash screen feature. These efforts added improvements by standardising the source code, as well as keeping the user better informed about the start-up

progress of the VCS. Unlike the old implementation, the current VCS splash screen, as shown in Figure 5.7, is displayed as soon as the application is executed. Textual information is dynamically drawn over a section of the VCS image, providing the user with real time details on the progress of the application loading process (more details in Chapter 4, Section 4.2.3.6). The image loaded with the splash screen also provides information about the confidential nature of the VCS prototype and reminds the user not to distribute the software.

5.3.2. Introduction Window

Upon completion of the software initialisation processes, the splash screen is hidden and instantly followed by the introduction/set-up window (Figure 5.8).

This window is separated into the following sections:

- Welcome Text (label 'A' in Figure 5.8)
 - Informs about the preparatory steps in the following sections and reminds the user about the help file for new users (a HTML document viewer has been implemented, but no content has yet been written for the help files. This is an area that will be addressed for future versions of the VCS).
- Student Reports (label 'B' in Figure 5.8)
 - Provides a link that launches a new window that contains information about each student (see Section 5.3.3). It is recommended that new users inspect the reports prior to commencing the lesson.
- Simulation Settings
 - The user must provide the following before the lesson may commence:
 - * Lesson duration (label 'C' in Figure 5.8): user to select from a drop-down list with options of 5, 10, 15, 20, 25 or 30 minutes duration. The two main reasons why lesson duration is fixed are; 1) it better mirrors time constraints in the real world and 2) if the VCS is run in multiple groups, the fixed duration ensures that all groups complete the activity at the same time.

Chapter 5. Graphical User Interface

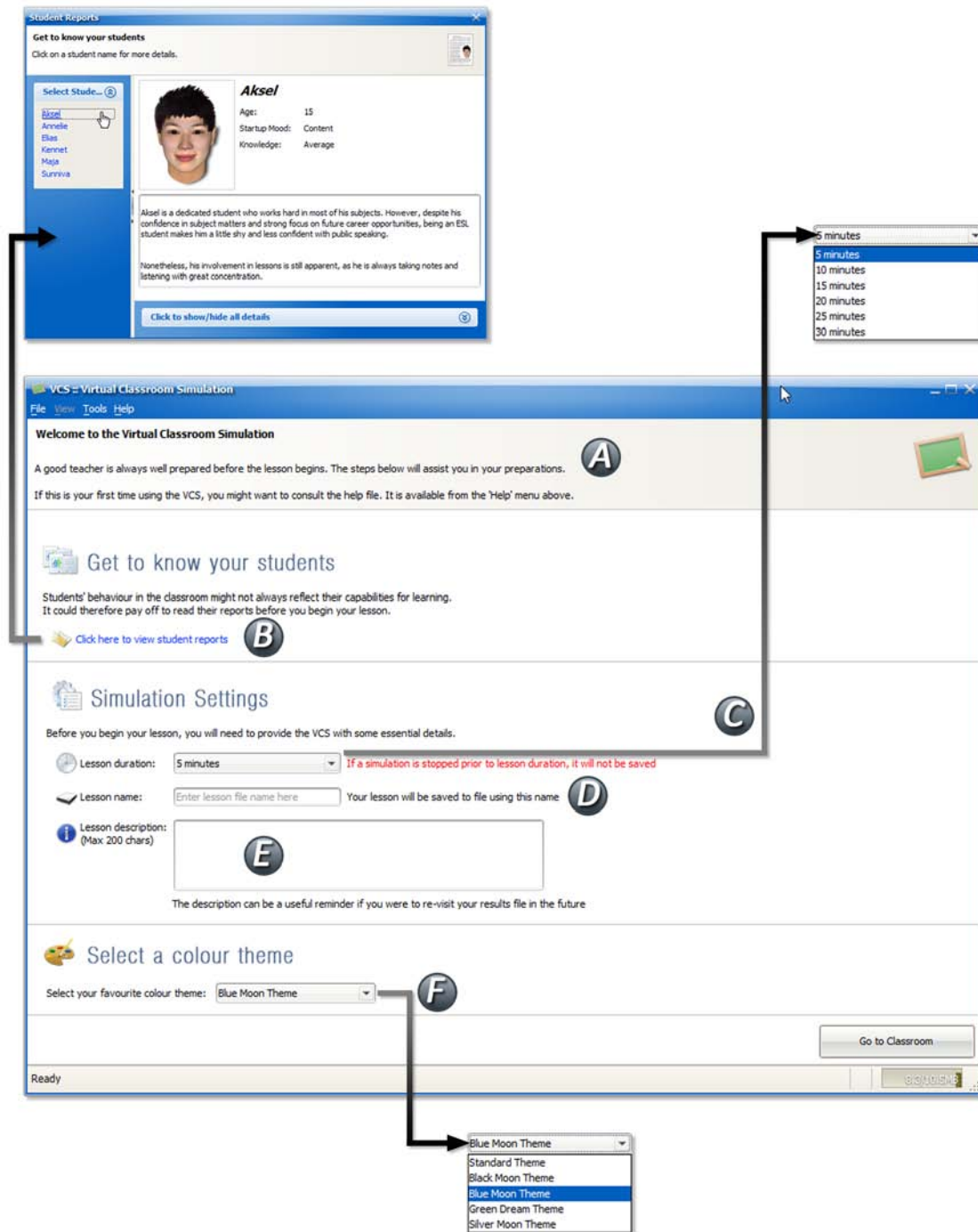


Figure 5.8.: VCS Introduction Window

Algorithm 5.1 VCS Session File (‘.vcs’)

```

#VCS Database File - DO NOT EDIT
#Mon Jun 15 02:16:21 CST 2009
description=A 5 minute lesson where Maja will be ignored
    throughout to observe how this may affect her mood and self-
    efficacy.
relative=vcsdb_15_06_2009_22_15_06
duration=300
absolute=C:\\Users\\simon\\vcs_data\\vcs_saved_sims\\
    vcsdb_15_06_2009_22_15_06

```

- * Lesson name (label 'D' in Figure 5.8): used as name for the file in which simulation settings are stored on the host computer file system. The user is prompted to enter a new name if it is identical to that of a previously stored simulation.
- * Lesson description (label 'E' in Figure 5.8): an optional short description of the lesson.
- Colour Theme (label 'F' in Figure 5.8)
 - A look-and-feel (‘cosmetic’) option that allows the user to choose between 5 different themes.

Simulation settings are saved in a separate file (extension ‘.vcs’) relative to where the database files are stored. The content of the file follows a **key=value** format, the content of which is seen in Algorithm 5.1.

The storage of this information is part of an (incomplete) implementation of a results viewer. This is a separate application that enables the user to view charts from past simulation runs. Two paths to the database root folder (relative and absolute) are provided to ensure that the content may be found even if the session file (‘.vcs’) is moved to a different location on the system. The description field is useful, because it specifies the simulation run in more detail prior to loading an entire data set into the results viewer.

The introduction window follows an itemised and well described structure that is easy to follow for the user. The design minimises the chance of user input error by implementing drop-down menus and imposing ‘sanity’ checks on mandatory fields (e.g. if a lesson name is not provided, or has already been used, the user is alerted).

5.3.3. Student Report Window

The student report window (Figure 5.9), which may be accessed either from the introduction window or the results window, contains a separate page for each student. The report includes readily visible information such as a student photo, description, mood and knowledge. Users are encouraged to read these reports prior to the lesson in order to get to know their students. While all other start-up attribute values may also be viewed here, they are hidden by default. The rationale for hiding this information is to keep the experience somewhat akin to the experience in the real world. However, *the option* to view student details provides the users with an opportunity to explore and to further understand the theoretical aspects behind the simulation.

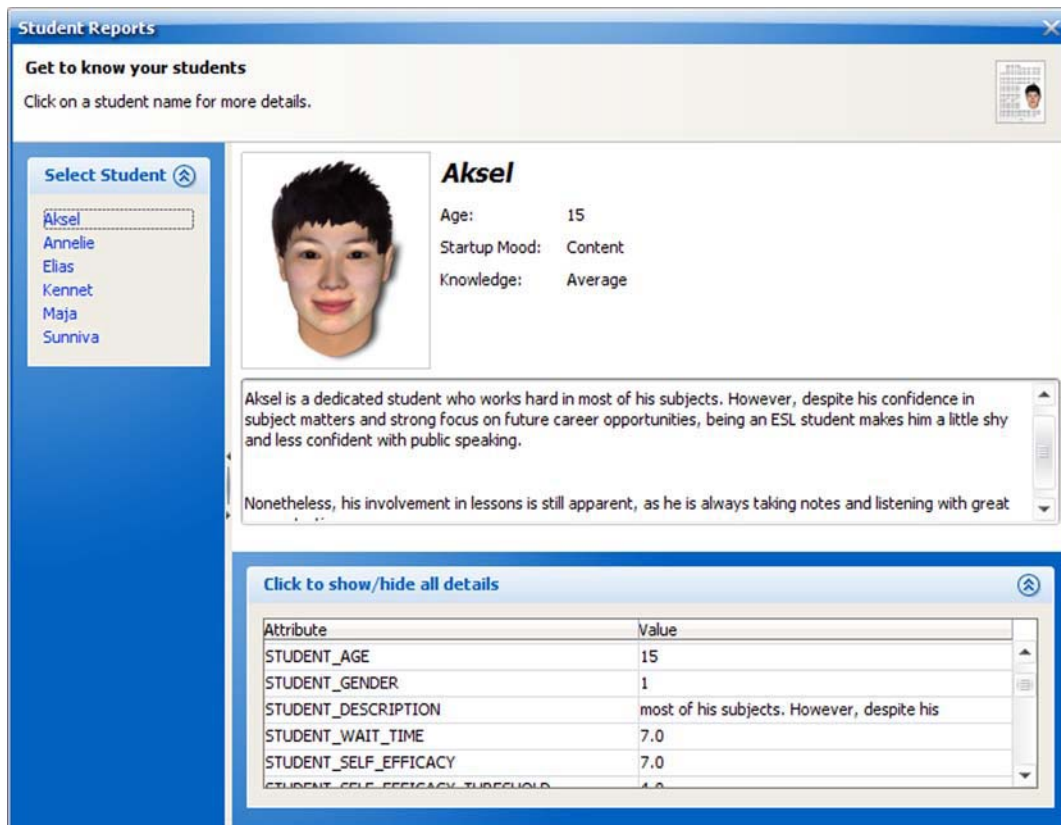


Figure 5.9.: Student Report Window

5.3.4. Main Window

When the user clicks on the ‘Go to Classroom’ button in the introduction screen, the window is re-populated with components of the main window. A 3-panel layout, which was originally introduced in the first series of rapid GUI prototypes, was employed to segment the different views of this window (Figures 5.10 and 5.12 on page 120).

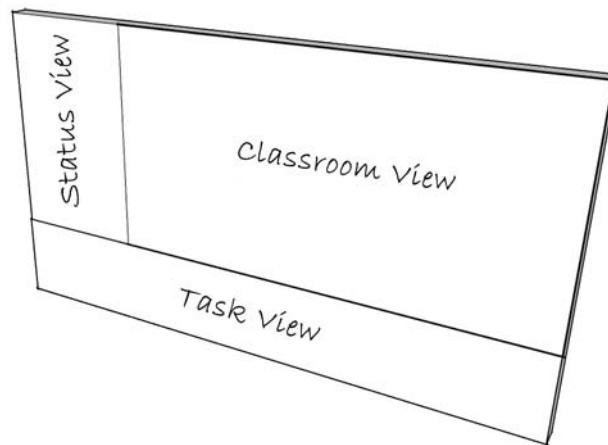


Figure 5.10.: 3-Panel View of the Main Window

The views, which will be discussed in more detail later, include the Status View, Task View and Classroom View. The layout of these views is realised using *FlexDock*, a Java Swing windowing and docking framework (Schaefer and Delap, 2007). Features of the framework that are utilised in the current version of the VCS include layout persistence, re-sizable containers with title bars and collapsible containers for saving screen real estate.

In this research, the term ‘real estate’ refers to the amount of space available for the VCS on the host computer’s display. While effective use of available real estate is very important, it can also be very challenging. The VCS’ main window contains a lot of information, textual as well as graphical, that must be organised in an intuitive way. The FlexDock framework was particularly useful in the GUI implementation because it allows for the segmentation, layout and resizing of views. The framework also allows for the prioritisation of real estate allocation, which in the VCS is given to the classroom view. Thus, when resizing the main

window, the real estate allocation for the classroom view will increase/decrease while the other two views keep their current size. Several frameworks were considered for the 3-panel layout, but in the end FlexDock was chosen for its features, ease of adaptation and aesthetics.

Most textual and graphical components of the VCS do not scale (e.g. their sizes are fixed and only the spacing between them may shrink or grow when the main window is re-sized). Thus, the totality of space required for all components imposes further real estate related issues on the host computer's display. In an effort to address this, a number of GUI prototypes were developed in the current research study. The views and the components contained therein had to be carefully tailored, not only to fit inside the window, but also to allow for dynamic placement when re-sized. Due to the abovementioned scaling issues, a decision had to be made with regard to the minimum window size (the smallest size possible before components no longer fit in their views). The current version therefore enforces a minimum screen resolution of 1024x768 pixels. If the host computer's display does not meet this requirement, a warning dialog is displayed to inform the user on application start-up (see Figure 5.11). The term 'dialog' is used in this context to describe pop-up windows that overlay other windows (e.g. warnings, errors, user input, information and help).

5.3.4.1. Classroom View

Before the user clicks the 'Start' button in the toolbar (which initiates the lesson), the entire classroom view is occupied by a graphic. This sketch, as shown in Figure 5.12, provides a visual representation of the classroom.

Its intention is to give the user a preliminary conception of the classroom layout before the simulation commences. The researcher's artistic skills are arguably very limited, but the free three-dimensional (3D) sketching software 'SketchUp' by Google (<http://sketchup.google.com/>) provided the means to create something that resembles a classroom.

The classroom view hosts six sub-panels laid out in a 3x2 grid; one panel per student. Each individual student panel, two of which are shown in Figure 5.13.

Components included in a student panel, their use and purpose are listed below:

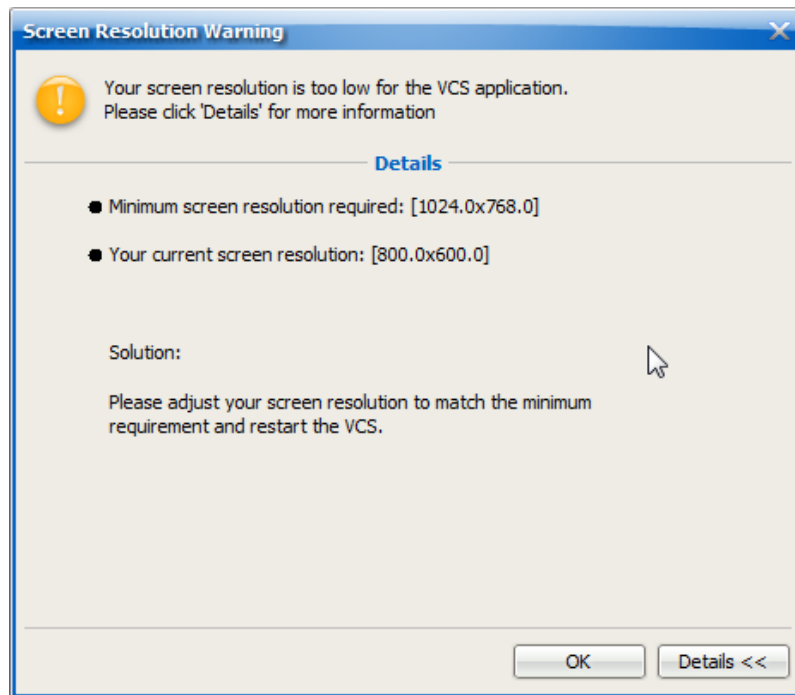
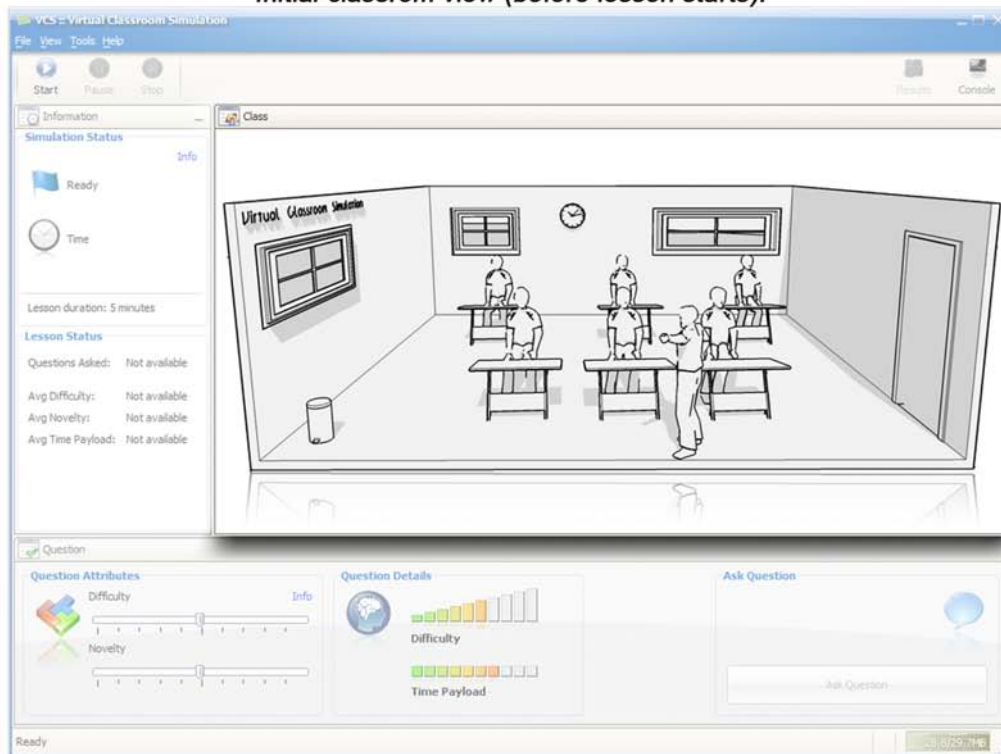


Figure 5.11.: Screen Resolution Warning

- A distractibility meter provides a continuously updated vertical bar to indicate distractibility of the student. The bar is colour coded from green (safe level) through to red (too distracted). Hovering the mouse over the bar displays a tool tip with further information.
- A student image is updated whenever student mood changes. Hovering the mouse over the image area displays a tool tip with further information.
- A ‘get attention’ action implemented as a clickable link is enabled when a student’s distractibility reaches a critical level.
- An ‘ask student’ action implemented as a clickable link is enabled after the teacher has asked a question to the class. Clicking the link will direct the selected student to answer the question.
- An animation with a ‘thinking’ label is displayed (in the top right corner) only when a question is asked to the class. If the student is too distracted the animation will not appear.

Initial classrom view (before lesson starts):



Classrom view when lesson is running:

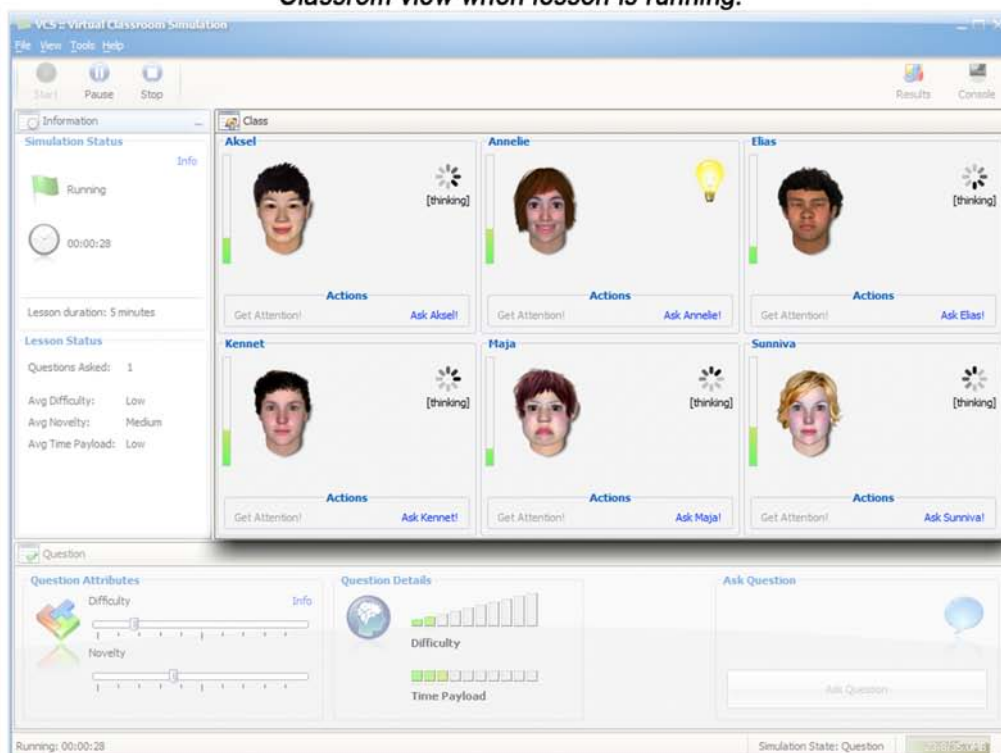


Figure 5.12.: Classroom Views Before and After Lesson Start

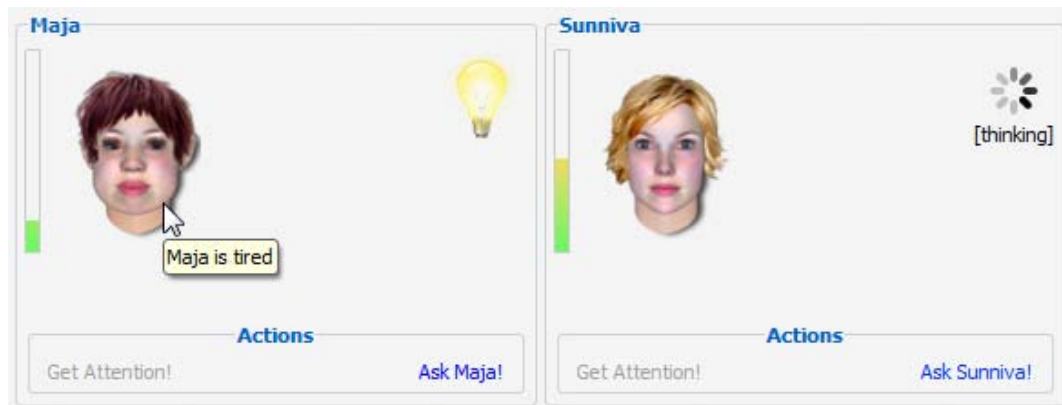


Figure 5.13.: Student Panels

- A light bulb animation signifies that a student is raising his/her hand and is ready to answer the question. The animation will appear only if the student is willing to answer the question.

Thus, some components in the student panels are exclusively controlled by the simulation framework (specifically the owning student instance) while others are modifiable through user interaction.

Dynamic placement of GUI components is achieved by nested panels, each of which with a manager most suitable for the required layout. When the classroom view is re-sized, the layout managers ensure that the student panels, belonging sub-panels and components are placed relative to each other.

5.3.4.2. Status View

Although the status view implemented in the Honours version (see Section 5.2.2.1) was very basic and provided little information, it nonetheless provided opportunities for a re-think of some core issues of the simulation. In the efforts of mimicking a real-world classroom, would it really be appropriate to display statistics about each student? How much information should be given away, and in what ways would this affect the type of learning and the overall ‘realness’ of the simulation?

Hindle (2002) highlights the importance in education for empathy of the ‘real-life’ situation. Feinstein and Cannon (2002) discuss in detail the concept of simulation fidelity (e.g. a simulation’s level of realism) and how it may impact on learning.

They point out that “a simulator does not need to be an exact representation of the real world to provide effective training” (Feinstein and Cannon, 2002, p. 427). Although the VCS cannot represent an exact replica of the real world, certain aspects of the simulation aim to maintain a close link to the realness of what is represented in an actual classroom environment. In consultation with supervisors, it was decided to take away most student-specific statistics so as to better reconstruct and represent a real environment. The information and visual composition of the view has therefore undergone major changes from earlier work.

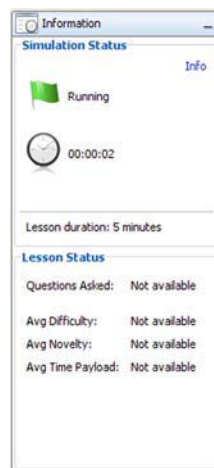


Figure 5.14.: Status View

The current status view (see Fig. 5.14) does include some details about the questions, e.g. a counter of how many questions have been asked in total and the question difficulty, novelty and time payload (approximate time it takes to ask a question) as averages. However, these averages are given only as textual value indicators, as opposed to real numbers:

- “Low” = Difficulty/Novelty/Time Payload averages < 30%
- “Medium” = Difficulty/Novelty/Time Payload averages between 30% and 70%
- “High” = Difficulty/Novelty/Time Payload averages > 70%

In the event that this highly abstracted information is still too detailed for a particular activity in an education course, the FlexDock implementation allows the status view container to be collapsed (e.g. hidden).

Information about the simulation status (Ready | Running | Paused | Stopped), simulation time and lesson length has also been added to the status view in this version. Icons have been added to supplement textual information, e.g. different coloured flags (🚩) are utilised to visually re-enforce simulation status (Ready = Blue | Running = Green | Paused = Yellow | Stopped = Red). Some information is also mirrored in the status bar (bottom bar of the main window), which includes simulation status, time and simulation state (Idle, Question or Interaction).

Included in the status panel is an 'Info' link (top right corner in Fig. 5.14) which provides more information/help about the items in the view, specifically:

This panel contains details about the simulation.

Status Flag

- Indicates what state the simulation is in:
 - Ready: Simulation has not started yet
 - * Running: Simulation is running
 - * Paused: Simulation is paused
 - * Stopped: Simulation has been stopped and cannot be resumed

Time

- Indicates how long the simulation has run for
 - Time is only accumulated whilst in the 'running' state

Lesson Duration

- This is how long the simulation will run for
 - Note: Actual running time might be longer (simulation will only end whilst in idle state)

5.3.4.3. Task View

As detailed in Chapter 3, Section 3.4.3.2, the task entity and the concept behind it have undergone some drastic changes in this research. A strong emphasis was placed on the ease of use and an attractive visual implementation of the task view (Figure 5.15). To this end, the Painter API (`org.jdesktop.swingx.painter`), included in the SwingX project (Sun Microsystems Inc. & Contributors, 2007), was used to achieve reflection of icons and background gradient effects in the task view.



Figure 5.15.: Task View

The view is segmented into three panels. The first ('Question Attributes') contains two slider components for setting question difficulty and novelty. The second panel ('Question Details'), which is non-interactive, provides a visual representation of the question difficulty and estimated time payload on a scale from 1-10. The third panel ('Ask Question') contains a button which, when clicked, activates the 'ask question' event. This event effectively broadcasts the user-composed question to all students. More information about the implementation of the task entity and its interaction with the simulation framework can be found in Chapter 4 (Sections 4.2.3.3 and 4.2.3.7).

Similar to the status view, the task view also includes an 'Info' link that provides the following information about this particular view:

Use the sliders to change the properties (difficulty and novelty) of the question attributes and observe how they impact on the time payload displayed in the Question Details panel.

Difficulty

- This adjusts the difficulty of a question.
 - Remember that an easy question generally takes less time to ask than a difficult one.

Novelty

- The novelty refers to the interestingness of the question. Any question can be made more ‘interesting’ in its delivery, but it will take more time to deliver.
 - While it may be tempting to always ask high-novelty questions, beware of the increase in delivery time (time payload).

Time Payload

- The time payload indicates approximately how long it will take to ask your question. As you gain more experience with the VCS, you will be able to tell (roughly) the correlation between the payload scale and actual time payload.

Keep in mind that your students might have different preferences when it comes to question difficulty and novelty; In order to please everyone, you might have to vary your question settings frequently.

A layout manager ensures that the first two sub-panels of the view (‘Question Attributes’ and ‘Question Details’) are kept at a fixed width and location (bottom left). Placement of the third panel (‘Ask Question’) is dynamically determined based on window size, always kept in the bottom right corner.

5.3.5. Results Window

At the end of a lesson, the VCS automatically loads and displays a results window which contains an interactive representation of the lesson history coupled with reports on critical simulation events. The service intends to enable users to make

connections between theory, practice, experience and reflection and, hence, the praxis of teaching.

While the planning and implementation of a results service was offered very little attention in earlier work, it was a pivotal activity in the current study. Understanding not only what to capture, but also how to logically store and retrieve simulation data, had to be carefully considered. This was particularly important since the GUI for this service was one of the last modules to be implemented.

Although simulation results can be presented in a number of ways, the intention was always to avoid plain text records. It was, however, anticipated that implementing a graphical representation of results would be difficult and quite possibly too complex to achieve in the current study. A worst-case scenario of plain text records was therefore a possibility. The best-case scenario, on the other hand, was to achieve an interactive graphical representation for each student inclusive of a timeline, variable attributes and events.

In the later stages of VCS development, a number of charting and reporting modules for Java were trialled. Due to its extensive features, ease of implementation, available examples and documentation, JFreeChart (Gilbert, 2007) was quickly adopted for the VCS results service. In fact, the features and extensibility of JFreeChart superseded the study's projected best-case scenario. The integration of the JFreeChart component, samples of which are shown in Figures 5.16 and 5.17, delivers the following features:

- Timeline on x-axis, attribute value on y-axis
- Student attribute plots (distractibility, metacognition, knowledge state and self efficacy)
- Simulation event states as coloured (and labelled) segments on chart background
- Simulation event details as interactive (left-clickable) hotspots
- Zoom, print and export to image file

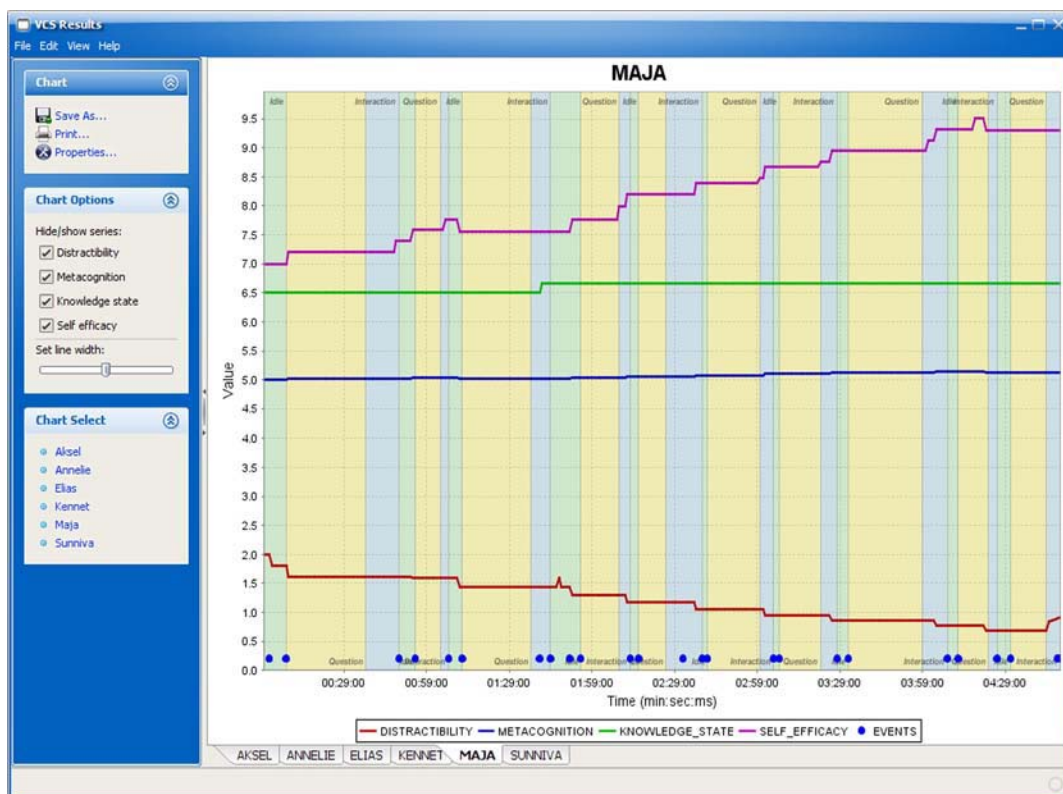


Figure 5.16.: Results Window

A right-click on a chart brings up a menu with options for zoom, print, save (to image file) and chart properties (in which labels, plots and colours may be altered). Most of these options may also be accessed from the main menu and the custom built side menu panel. The side menu also provides means to hide/show attribute plots and to change their width.

Configuring the JFreeChart component to draw the plots was straightforward, mainly thanks to the way data are managed in the VCS and the ease of which data result sets can be extracted and manipulated with the Structured Query Language (SQL). To configure the axes (e.g. labels for time format and flexible attribute values) and legend layout (e.g. positioning and shapes) was not complex, though it demanded some tinkering. Implementation of the chart background, which indicates periods of event states, was perhaps the most demanding activity since it required a more customised solution. JFreeChart's support documentation and the Application

Programming Interface (API) was therefore a very useful resource to have at hand.

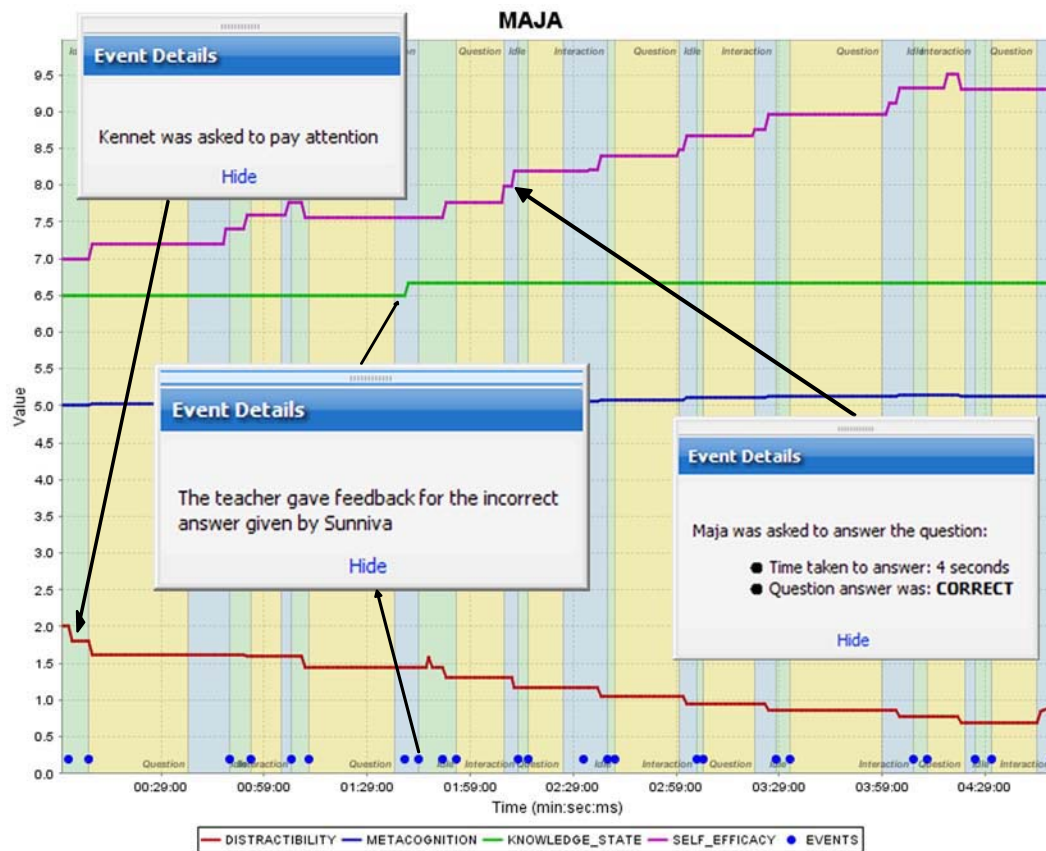


Figure 5.17.: Student Chart (Maja) Explained

Another important feature built into the charts, though not explicitly offered by JFreeChart, is the ability to left-click on an event marker to display further information about that event. This was achieved by implementing the mouse listener interface provided by the JFreeChart component in order to determine where the event markers are positioned on the chart. Each marker has a unique ID which is linked to its record entry in the events table of the simulation database. Thus, when a left-click event is caught on an event marker a pop-up window containing information from the marker's corresponding record entry is displayed (see examples in Figure 5.17). Rather than developing a custom solution for the pop-up window (utilising a JPanel with added sub-panels for dragging, information display and a close button), a pop-up component developed by JIDE Software (part of the JIDE

Common Layer, an Open Source Project) was adapted for this purpose.

The results window includes a help dialog which is accessible from the **Help** menu. The help document, content of which is included in Appendix F, provides a detailed account of how to read and interact with the charts. Also provided is a complete record of the event table, which can be accessed from the **View** menu (see Figure 5.18). This tabulated representation includes information additional to what is accessible in the charts, e.g. time and date of simulation start, pause, resume and stop.

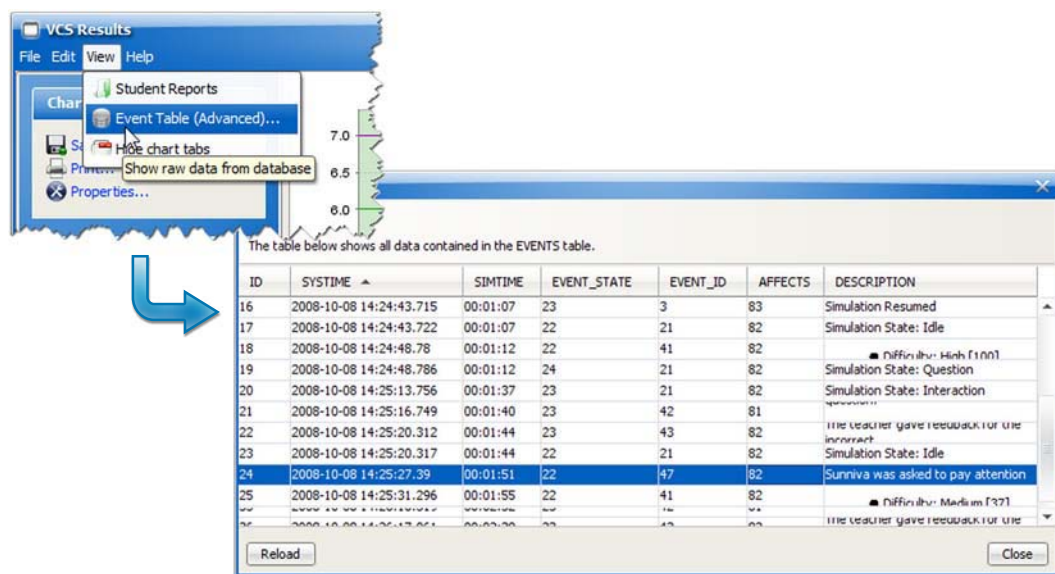


Figure 5.18.: Menu Access for the Event Table

Although the results are displayed automatically at the end of a lesson, the user may also open the results window whilst the lesson is running. In this case the simulation is automatically paused and controls in the main window are blocked from user interaction.

5.3.6. About Dialog

A ‘dialog’ (American spelling of the term intentional) is a popup window that overlays other windows in an application. Dialogs are commonly used for program alerts, information and other user interaction. As with most other software, the

VCS provides an about dialog which can be accessed from the **Help** menu (see Figure 5.19).

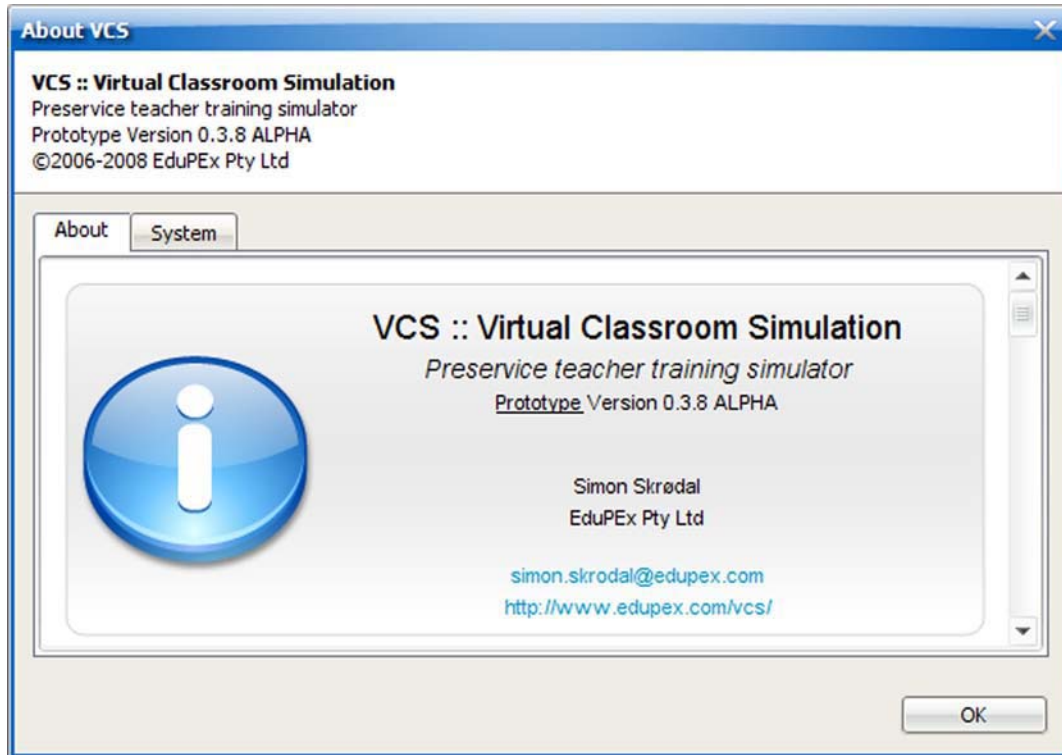


Figure 5.19.: About Dialog

The ‘About’ tab, which displays a hypertext document (`about.html`), provides copyright information, software version number, contact details and credits. In order to make software updates more automated, the hypertext document is dynamic in that it includes a number of custom tags (placeholders) that are replaced by the VCS software at run-time (see Table 5.1).

This eliminates the need of having to edit the hypertext document in the event of version upgrades of the VCS and other core 3rd party components. The full content of `about.html` (as displayed by the about dialog) is available in Appendix G. The ‘System’ tab, as the name suggests, provides extensive information about the host system (including Java version and paths, details on operating system, path to user’s home directory and path to VCS databases). The dialog is an implementation of a component developed by Jyloo Software as part of the *SyntheticAddons* package.

Table 5.1.: Custom tags added to `about.html`

Tag	Description
{title}	Software title; defined in the VCS application resource file
{description}	Software description; defined in the VCS application resource file
{major}.{minor}.{revision}	Software version numbers; defined in the VCS application resource file
{developer}	Software developer; defined in the VCS application resource file
{vendor}	Software vendor; defined in the VCS application resource file
{year}	Software development year(s); defined in the VCS application resource file
{syntheticaVersion}	Synthetica Version; extracted from a Synthetica API method call
{addonsVersion}	SyntheticaAddons Version; extracted from a SyntheticaAddons API method call
{lafName}.{lafVersion}	Synthetica Look and Feel name and version; extracted from a Synthetica API method call
{derbyName}	JavaDB Product Name; extracted from a JavaDB (a.k.a. Apache Derby) API method call
{derbyVersion}	JavaDB version number; extracted from a JavaDB (a.k.a. Apache Derby) API method call

The look and feel used for the VCS, *Synthetica*, is also provided by Jyloo Software (2007).

5.4. Graphical Representation of the Classroom and Students

Working out the best achievable way of how to graphically represent a classroom with individual students was a major challenge, both conceptually and practically. A number of complexities had to be addressed in order to reach the most viable implementation in terms of what layout, interaction, visual indicators and optimise for available screen real estate. This is best explained by starting at the most desirable outcome and work backwards until reaching the state of the current implementation.

5.4.1. Desired Representation

The implementation of animated 3 dimensional models to represent VCS students (as opposed to the flat static images used in the current version) would likely improve the level of realism (also referred to as ‘fidelity’, Feinstein and Cannon, 2002) presented to the learner. The challenge here is the design of individual student characters, their location in classroom and how to programmatically control context-sensitive animation. Complex tasks, such as getting characters to scale and modelling body and facial features, would need to meet requirements of the graphical and grammatical components required for modelling movement, expression and, perhaps, speech. While many of today’s modern computer games achieve incredible details of body movement and facial expressions, time and resources did not allow such elaborate features for this research study. Instead, alternative and more feasible solutions had to be considered.

5.4.2. Challenges

Although the researcher was not able to design a graphical representation close to that described in Section 5.4.1, the importance of an appealing and informative

5.4. Graphical Representation of the Classroom and Students

classroom representation was nonetheless vital. Limited skills in graphical and 3D modelling techniques would prove to make this a major, not to mention time consuming, challenge.

Several 3D modelling and animation tools were studied, and some trialled (including AC3D, Blender, Maya, LightWave 3D, Autodesk MotionBuilder and SoftImage Face Robot). As the difficulty in building and animating complete human models using these tools became even more apparent, it also highlighted some other concerns. Many, if not most, of these software packages are feature-limited evaluation versions only, with the full versions being quite expensive. Further, there are often interdependency issues when using these tools; i.e. you may use one tool to create and export your models to a selection of more or less standardised formats. These formats may or may not be compatible with other tools that facilitate animation. Then there is the question of how the developed models can be controlled programmatically by Java code to produce dynamic and situational animations that represent human behaviour.

Eventually, it became apparent that animated 3D models would not be a feature offered in the VCS at this stage. An early prototype developed in 2003 utilised static images to represent student mood. The idea was to have a collection of unique student faces, with each student having a subset of facial expressions that would be displayed to represent any given current mood state.



Figure 5.20.: Student Representation in VCS Prototype 2003

As seen in Figure 5.20, the image size (19x19 pixels), detail (cartoon-like) and quality (basic set of colours), as well as the number of unique faces (3 students) and mood subsets (happy, normal and angry/upset) were limited in this version. However, the notion of using static faces was useful. Writing code that enables the alternation between static images for reflecting changes in student mood is

straightforward. More challenging is the design of good quality graphics for each student. In order to achieve this, possible options were:

1. Search for readymade non-proprietary images, preferably computer generated, of ~15 year old individuals expressing a range of moods.
2. Find a software application that allowed for straightforward generation and manipulation of 3D face models.
3. Pay someone to produce the graphics.
4. Get permission from a group of selected students (as well as their parents and the University Research Ethics and Compliance Unit) for the purpose of photographing them and using these images in the software.

Unfortunately, no image sets were found to suit desired criteria for the VCS student representation. Similarly, initial search for suitable software only produced references to image manipulation software and complex 3D modelling tools already trialled and deemed unsuitable. For several reasons, it was difficult to resort to options 3 and 4 in order to get around the problem, so quite some time was spent on finding photographs and drawings online that could potentially serve as templates for image-manipulation. In the end, no viable solution was found, and the issues of graphical student representation were placed on hold whilst working on other areas of the project.

5.4.3. FaceGen

Following an extensive product search online FaceGen Modeller (Singular Inversions, 2007), a parametric face modelling software, was discovered. The software, which had reached version 3.1 at the time of use, included a number of features that matched the requirements of the project. FaceGen is not free software, though a functional evaluation version is available for download.

With FaceGen Modeller you may generate your own 3D faces by loading one or more photographs into the application. A few 3D faces are included with the installation, some of which were used in the generation of the VCS students. A face can be tweaked in a number of ways in order to change its appearance, including

age, gender and race morphing, as well as over 50 symmetric controls and a number of skin textures.

Six unique student faces, each with a subset of six mood states, were created using FaceGen Modeller. Student ‘Elias’ can be seen in Figure 5.21. Similar representation of all students can be found in Appendix B.



Figure 5.21.: Student Faces Generated With FaceGen Modeller

The evaluation version of FaceGen Modeller imposes some limitations; e.g. save and export features are disabled and a watermark logo is added to the forehead on the model’s preview image. Further, the software has no provision for adding hair; a separate software, ‘FaceGen Customizer’, is needed to do this. The export limitation was resolved by taking screen-shots of the face previews and saving these to image files. Although the image resolution would have been better had they been exported with the full version of the software, the screen-shots were more than adequate for their purpose. Hair from random photographs found online was added to the models using the graphics editing program Photoshop. Incidentally, this also hid the watermark as the superimposed hair covered the logo.

5.5. Third Party Technologies

It is important to note that several 3rd party technologies were brought together to build the simulation framework and the graphical user interface. The decision to use only open source (or otherwise free) components sometimes restricted what could be achieved in this regard. The VCS was developed over a prolonged period of time and a number of components went through incremental version upgrades in this time. These upgrades typically added extra features, bug fixes and other improvements of benefit to the VCS. At times, this meant that the VCS source code had to be modified to accommodate the use of upgraded components (e.g. when methods in a component’s API become deprecated).

The researcher was active in reporting bugs and feature requests to developers, which would typically lead to new and improved versions being made available quite promptly. Due to the extensive involvement with the Synthetica and SyntheticaAddons products from Jyloo Software (2007), the researcher was granted extra access privileges to some of their commercially licensed components.

Appendix A provides a complete list in which third party technologies utilised in this study are acknowledged.

5.6. Summary

An efficient, user-centred, design of the GUI needed to take a number of constituents into consideration. The main objective of the VCS GUI is to serve as a communication layer, a layer of abstraction, between the user and the simulation framework. As such, development of the simulation framework and the GUI was frequently done in parallel.

Although GUI design of the classroom was partly conceptualised in earlier work, it was vastly improved in this study both with regards to functionality and aesthetics. Rapid prototyping and a countless number of refinements were involved to achieve this. Every facet of the graphical interface was designed with the end user and usability in mind. Constraints were put in place to ensure that the user cannot do ‘the wrong thing’ (e.g. error checking of user input, blocking of user input in special cases, message and alert dialogs, integrated help/information, etc.). In order to make the interface ‘cleaner’ and easier to observe, it makes extensive use of icons and other visual indicators. Further, information and actions are grouped in a logically segmented layout. The user interface follows a standardised design that users would be familiar with from other computer software (e.g. splash screen, menu/tool/status bars).

The user interface is tightly integrated with the simulation framework. This is achieved through a sophisticated model of events and event handling. The GUI and the framework were both developed with future scalability in mind. Thus, adding new features to the simulation (framework and GUI) is straightforward due to the modular nature of the current implementation.

What was achieved in the design and development of the GUI pushed the

envelope of the expected outcome. Nonetheless, a number of limitations are apparent. Specifically, the static representation of VCS students limits the realistic expression of body language, facial expressions, distractibility and so forth. While a tangible alternative solution was achieved, student animation will be given a high priority in future development.

The research design of this study puts measures in place to gauge the usability of the VCS (determined by the GUI and software performance) as perceived by the population of education students. This will be described in more detail throughout the remaining chapters.

Chapter 6.

Research Design

Numbers are useful, but they have to be meaningful for appropriate interpretation.

(ALAGUMALAI, 2008)

Chapter 2 specified the aims and objectives behind the application of the VCS. These are complemented by this study's research problems, which were assembled as a device to help establish if a simulation of this nature may indeed have the capacity to address said aims and objectives. A set of methods were mindfully put together as part of a research design that collectively sought to respond to these problems. Feedback from the target population (i.e. education students) was given high priority and several measures were put in place to facilitate the adept collection and processing of information. Data of both qualitative and quantitative nature were collected and analysed using techniques from both the qualitative and quantitative paradigms. This mixed methods approach, as visualised in Figure 6.1, will be substantiated further in this and later chapters.

The unique trans-disciplinary nature of the study, is, as indicated by the research problems, an intertwined conglomeration of educational theory, modelling of cognitive and non-cognitive entities, simulation framework, user interface design and advanced measurement and evaluation research methods. Therefore, a variety of methods had to be implemented for the capture and analysis of both qualitative and quantitative data to suit the intricate nature of the research. Some of the captured information was also converted to fit with other datasets (e.g. qualitative to quantitative), which provided an even higher level of detail in the data analyses.

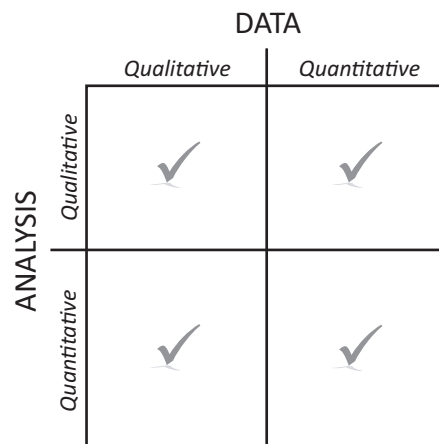


Figure 6.1.: Mixed Methods Approach of this Study (adapted from Hurmerinta-Peltomäki and Nummela, 2006, p. 445)

The research design was constructed through relevant literature, well-established instruments and supervisory advice. Throughout the candidature, the researcher audited a number of university courses (e.g. “Measurement, Evaluation & Assessment”, “Quantitative Educational Research”, “Educational Inquiry” and “Qualitative Research Methods”), all of which provided further insights into the skills and knowledge required to devise an appropriate design for this research.

6.1. Population

The target population (participants) of this study was described in detail in Section 1.4, Chapter 1. To recapitulate; students enrolled either in the 4th year Bachelor of Teaching or the Graduate Diploma in Education programme in semester 1, 2008 (see Table 6.1) were invited to participate in the study. The VCS was incorporated as an integral part of “Student-Teacher Interaction in the Classroom 1” (STIC), a compulsory course for these students.

Table 6.1.: Population and Gender Distribution

	BTeach	GradDipEd	Total
Female	38	79	117
Male	23	55	78
Total	61	134	195

6.2. Ethical Considerations

Prior to user trials and collection of data, appropriate ethical clearance from the Human Research Ethics Committee of the University of Adelaide (HREC) had to be obtained. Specifically, the collection process and the instruments to be used, which included survey items and the information sheet, were subject to approval. The research study *Virtual Classroom Simulation*, project number *H-136-2006*, was approved by the HREC on 11 October 2006 (see Appendix H).

6.3. Methods

Inherent in the research problems (discussed in Chapter 1, Section 1.3) are a number of questions that need to be answered; e.g. with regard to software usability (technical), validity of simulation output (theory) and knowledge transfer (learning). In order to address these problems, this study implemented a pedagogical framework consisting of a number of sequentially connected activities (questionnaire, lecture, trials, surveys, and an assignment). The order in which these activities were arranged can be observed in Figures 6.2 and 6.3, while Figure 6.4 provides an overview of the surveys employed in this study.

Specific details about the research methods and activities are provided throughout this and subsequent chapters.

6.3.1. STIC Questionnaire

An important aspect of teacher training that the VCS aims to address is the capacity/opportunity to reflect and to investigate the various constructs involved

Chapter 6. Research Design

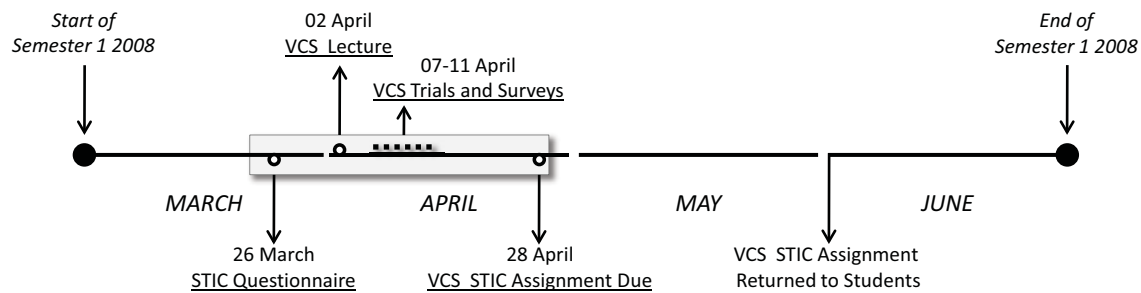


Figure 6.2.: Timeframe of the Population's Involvement with the VCS

in classroom interaction. In essence, this can be narrowed down to what is sought after in Research Problem 1 of this study (e.g. the most relevant entities in a classroom environment, their most significant attributes and how they interact). As a means to indirectly introduce this concept to the participants prior to them having any knowledge of the VCS, a reflective exercise was introduced. In this first point of contact with the participants, all students in the STIC 1 course were asked by the course coordinator to complete a single-page questionnaire on student-teacher interaction (included in Appendix J). The three items in the questionnaire aimed to stimulate and gauge the students' thoughts about STIC and, indirectly, address the first research problem of this study. Specifically, the questionnaire items were:

1. *Briefly identify at least three characteristics of a teacher that would be useful in the interaction in the classroom?*
2. *What factors affect student learning in the classroom?*
3. *In posing a question or task to a student, what do you bear in mind about the question/task?*

Each question asked for at least 3 characteristics/factors to be included in the response. The questionnaire ascertains the education students' preconceptions and mental models of (1) the teacher, (2) classroom/student factors and (3) the task/question. Hence, it establishes the basis for a conceptual modelling exercise similar to what was undertaken by this study to address Research Problem 1. At the time of completing the questionnaire, however, the education students were

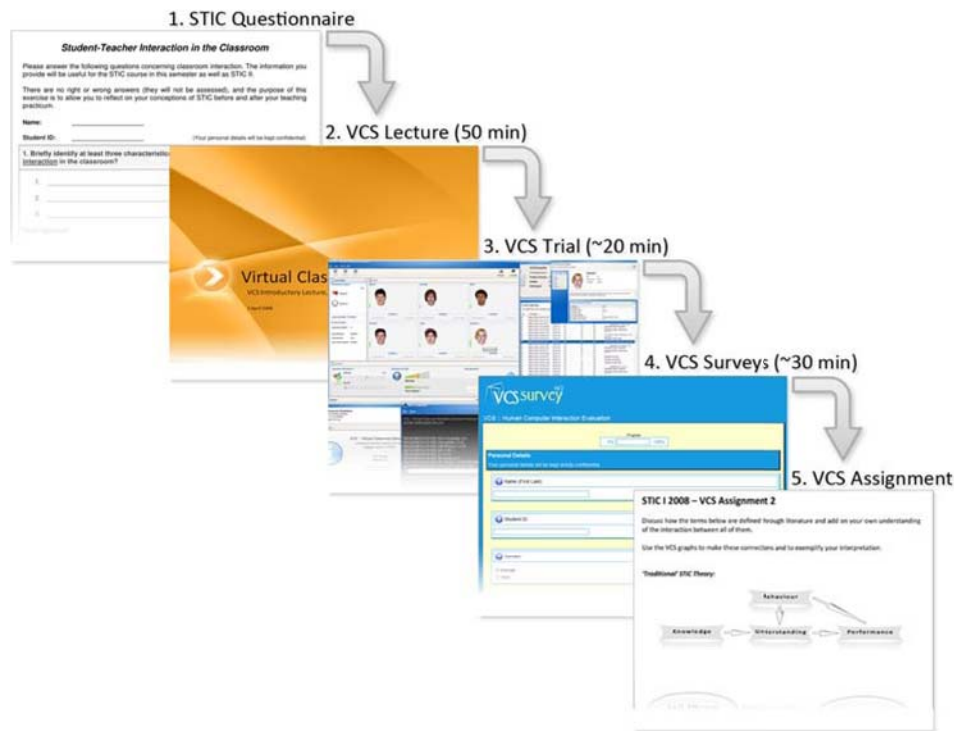


Figure 6.3.: Sequence of Activities Involving Participants

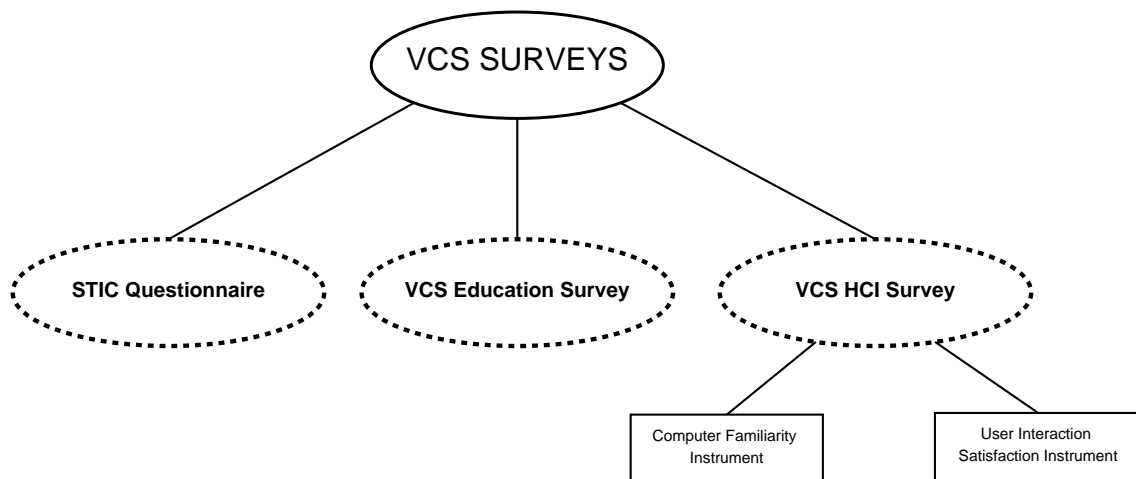


Figure 6.4.: VCS Surveys

still unaware of the VCS and the forthcoming introduction lecture. The responses were, as such, useful in providing an unbiased insight into the education students' thoughts about STIC as well as preparing for the introduction of the VCS.

Eighty four students completed this questionnaire and collected data were transcribed into an electronic format for further processing and later utilised in the VCS introductory lecture as discussed below.

6.3.2. VCS Introductory Lecture

A week prior to the VCS software user trials, the researcher held a 1 hour introductory session to all participants about STIC, simulations and the aims and objectives of the VCS. In this session, the researcher also made reference to the recently administered STIC questionnaire and revealed its association with the research study. In order to provide a quick and easy to read overview of the collected data, the information was collated into a "word cloud". Word clouds (or tag clouds) are commonly used in web documents to represent occurrences of words/tags used on a website (e.g. frequency of search words/terms). In this instance, the word cloud entailed a graphical representation of the frequency words used by the students in the questionnaire. The cloud included only the most commonly words used and, as seen in Figure 6.5, words used more often were also displayed with a larger font size.

This type of representation was a useful utility for demonstrating an overview of student responses. It evoked much attention by the students, who were now given the opportunity to reflect on how and where their individual responses fit in the cloud. It also allowed the researcher to establish parallels between the questionnaire and the aims and objectives of the VCS.

A handout describing the simulated VCS pupils in more detail (see Appendix K) and preparatory information for the following week's trials and surveys were distributed to the population at the end of the lecture. The complete set of slides presented in the session is available in Appendix L.

students student task environment question teacher
 knowledge good learning clear level ability classroom class
 interesting understanding skills .question task topic behaviour
 confidence enthusiasm relevant questions patience physical understand
 motivation time communication teaching interest Relevance prior engaging
 open learn voice engage answer resources humour friendly emotional subject
 background teacher's distractions Approachability attitude required .prior knowledge.
 cultural content concise age encouraging organised .classroom environment.
 management .communication skills. activities .emotional environment. .cultural
 background. .good communication. knowledgeable outcomes outcome abilities Home
 difficulty .task question. personal .students level. group set sense Peer interaction
 engagement teachers personality interactions discussion Listening Technology tasks
 organisation ways .learning environment. clarity individual listen material atmosphere
 approachable Interactive .listening skills. ended Compassion explain lack direct listener Empathy good

Figure 6.5.: STIC Questionnaire Word Cloud

6.3.3. VCS Trials

The VCS hands-on trials were administered over a single week, covering all nine tutorial time slots available to STIC 1 students (each student would attend one tutorial only). The tutorial time slots were determined by the course coordinators at the start of the semester. In the week prior to the trial, the VCS introductory lecture had already provided participants with information about its purpose and structure. At the start of each tutorial session, participants were provided with a Research Project Information Sheet (see Appendix M), a standard consent form for people who are participants in a research project (see Appendix N) and an instruction sheet for the trial session (see Appendix O). Consent forms were filled in by participants and collected by the researcher during the installation of the VCS software. Each tutorial was limited to 50 minutes only; a time slot that had to be shared with VCS software installation and two surveys (see Figure 6.6).

Software installation was a time consuming exercise that could not be avoided due to system configurations/restrictions imposed by the Information Technology Services at the University of Adelaide (see Appendix P for more details). As will be discussed later, the time available for the trials and surveys is a limitation of this study.

When the software had been installed, participants were asked to observe the

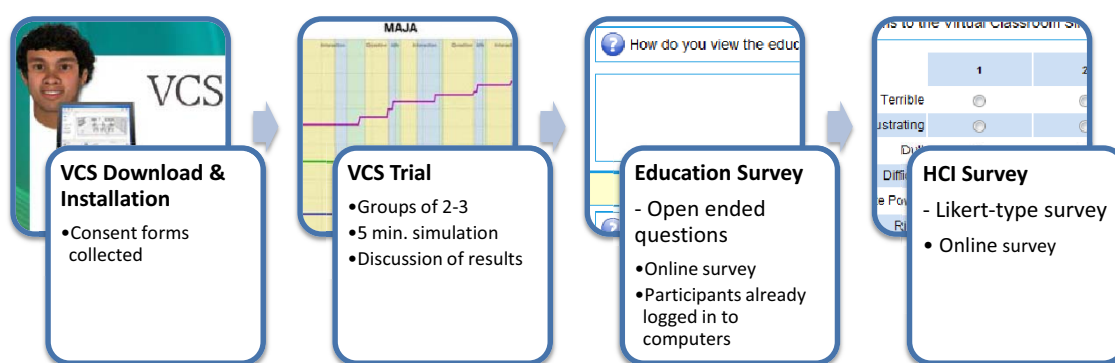




Figure 6.6.: VCS Trial: Sequence of Events

particulars of the VCS welcome screen (including student reports). They were then instructed to configure the VCS to run a 5-minute lesson and commence the simulation. Working in groups of two or three, participants were encouraged to interact, observe and discuss both during and after the simulation run. Grouping of participants was a design decision to promote discussion through critical thinking and reflection; outcomes that enable the participants to become more skilled (Osterman, 1990). Once the simulation was completed, the researcher provided details about the simulation output and demonstrated on a projector screen how to interact with the charts. A few minutes were then allocated for the groups to discuss their own simulation output (charts). This exercise allowed students to discuss their own understanding of the relevant theory into service and to explain possible reasons for the observed results. The trial was concluded with an open discussion involving all participants, allowing for any questions or comments to be shared with—and responded to by all participants.

At the end of the trial, participants were asked to individually respond to two online surveys. The VCS installation file included Internet shortcut files that pointed to their locations (i.e.  Survey 1 - Educational and  Survey 2 - HCI). Students were instructed to first do Survey 1 (“Education”, see Section 6.3.4) followed by Survey 2 (“HCI”, see Section 6.3.5).

Further details about the mode, rationale and administration of the data collection stage can be found in Section 6.4.

6.3.4. VCS Education Survey

The Education survey (see Table 6.2) takes into consideration the educational understanding and impact of the VCS; e.g. the successfulness of the training simulation in teaching and learning. It ties in with Research Problem 1 by gaining an insight into the education students' understanding of the most relevant entities in a classroom environment and what they see as the most significant attributes. Research Problem 2 is also addressed in this survey through questions concerning the value, strength and improvements of the VCS. Where the survey adds most value, however, is with regard to Research Problem 3; e.g. to what extent the VCS is useful in reshaping preservice teachers' mental model about teaching and learning, as well as advancing their pedagogical content knowledge. A number of questions in the survey aim to gauge the students' understanding of cause-and-effect scenarios and how well they relate to theory and the real world while others are directed towards the VCS and its value in teaching and learning.

The survey, which consists of 18 open-ended questions, was developed by the researcher in collaboration with the supervisors in education. More details about this survey can be found in Chapter 7.

6.3.5. VCS HCI Survey

The VCS HCI (Human-Computer Interaction) Survey is concerned with the population's experience, use and attitudes towards computer technology in general, as well as their perception of the VCS technology specifically. It is composed by two established instruments, namely the *Computer Familiarity Instrument from the PISA 2000 Study* (OECD, 2000; Adams and Wu, 2002, pp. 242-243) and the *Questionnaire for User Interaction Satisfaction* (QUIS, Chin et al., 1988).

The PISA instrument aims to measure general experience, use and attitudes towards computer technology, while QUIS is solely focused on the users' satisfaction with the VCS. Both instruments are primarily of a quantitative nature and include dichotomous and polychotomous items. The instruments in their original form (e.g. prior to adaptation for use in this study) may be observed in Figures 6.8 and 6.9.

Chapters 8 (Computer Familiarity Scale) and 9 (User Interaction Satisfaction

Table 6.2.: VCS Education Survey: Question Items

Question Item	Question Text
1	How do you view the educational value of the VCS?
2	Do you believe the VCS is an effective means of training? How and Why?
3	Do you believe the VCS generates a valuable learning experience? How and Why?
4	On which element/aspect of the simulation did you place most emphasis? Please comment.
5	From your understanding of the VCS, what are the attributes of the students that the teacher need to take into consideration when posing a task or question?
6	Can you list the attributes of a teacher, as presented either explicitly or otherwise in the VCS?
7	In posing a question or task to the class or a student, what does the teacher need to know to minimise distractibility and optimise learning?
8	What are the strengths of the VCS? Please give specific examples.
9	From your knowledge and experience in classrooms, does the simulation meaningfully resemble the real environment? Please comment.
10	What improvements need to be incorporated into the VCS as a teaching and/or training tool?
11	Explain how knowing about [the student's prior knowledge about the topic] is important for the teacher?
12	Is self-efficacy affected by question answering?
13	How could students' level of prior knowledge about a lesson topic be related to their level of distractibility?
14	Explain how consideration of the level of difficulty of a question is important for a teacher.
15	Annelie had a high level of distractibility after about 1 min into the lesson. Why do think her level of distractibility increased and stayed quite high for the rest of the lesson?
16	What is Kennet like as a student?
17	If Maja's face showed less sadness during the lesson, why would this be?
18	From your understanding of the VCS, in what scenario(s) is a student's Knowledge State affected?

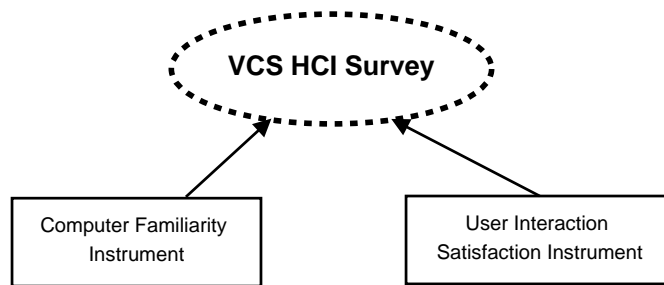


Figure 6.7.: VCS HCI Survey Composition

Scale) describe each instrument in detail, adaptations made for this study and the investigation of their validity and utility.

6.3.6. VCS Assignment

A student assignment with references to the VCS was designed by the course coordinator and the researcher (Appendix Q). This assignment, which made part of the formal assessment in the course, sought to extract students' understanding of terms from traditional STIC theory and how they might relate to specific VCS output (results). The submitted assignments provided further information relevant to Research Problem 3 of this study by allowing a more detailed analysis of the VCS' capacity to advance participants' pedagogical content knowledge.

The assignment included two student result graphs produced by the VCS (see Figures Q.2 and Q.3 in Appendix Q). The assignment asked that students use these graphs to exemplify their interpretation of the relevant theory. In what could be called a reflective exercise, students therefore needed to connect established theoretical concepts with their own experience from using the simulation, as well as the learning (change in mental models) that may have eventuated.

Results from this assignment, as marked by the course coordinator and one assistant lecturer, were taken into consideration in the data analyses.

6.4. Data Collection (VCS Surveys)

Data collection took place at the School of Education during Week 15, one week prior to the students' first practicum placement. All nine tutorial classes in which

NOTE:
This figure is included on page 150 of the print copy of
the thesis held in the University of Adelaide Library.

Figure 6.8.: Questionnaire for User Interaction Satisfaction v.5.0 (captured from
Chin et al., 1988, p. 215)

NOTE:
This figure is included on page 151 of the print copy of
the thesis held in the University of Adelaide Library.

the trials and data collection occurred were administered in person by the researcher. Care was made to ensure that all sessions followed the exact same structure so as to prevent any administrative inconsistencies.

6.4.1. Method of Collection

The fact that surveys were to be held in the same session as the VCS software trials (i.e. in a computer lab) presented an opportunity for electronic data collection. Employing an electronic mode of submitting data could, however, impose undesirable implications in terms of participant biases. In regard to response differentiation in electronic versus paper-based surveys, Boyer et al. (2002) provide evidence to suggest that they are generally consistent with each other. Similarly, a study which specifically involved an instrument used in this research found that the online format did not affect users' rating (Slaughter et al., 1994). In terms of positive/negative biases towards the use of an online survey system, it is already a requirement for the participating students to electronically compose and submit assignments, interact on electronic discussion boards and to access course content online. The general knowledge and acceptance of ICT in the VCS target population were therefore assessed to be sufficient for this method of collecting data.

Not having to manually transcribe hundreds of paper-based surveys of qualitative and quantitative data to a computer-readable format is arguably less time consuming, cheaper, makes administration easier and eliminates any potential errors in the paper-to-computer transcription process. Wright (2005), for example, highlighted the potential time and cost benefits afforded by commercial online surveys. Further to these advantages of online surveys, Boyer et al. (2002) also found reason to believe that the number of missing responses could potentially be reduced if data were collected electronically. The VCS surveys, for example, were configured to flag all items as compulsory, preventing respondents to skip items and submit incomplete forms.

Benefits of employing electronic data collection, however, are circumstantial and do not always outweigh the traditional paper-based mode (Fricker and Schonlau, 2002). Certainly, most of the aforementioned gains of using an electronic survey

rely heavily on aspects specific to the study in question and cannot be universally generalised. In particular for this research study, and contrary to exemplifiers in Boyer et al. (2002) (and Fricker and Schonlau (2002); Wright (2005)), the subject population is known and administered in a face-to-face environment; not by means of email invitations or random online surveys to capture certain segments of online communities. Further, the availability of free survey software suitable for this study was carefully investigated and deemed tangible. The researcher's technical capability to independently install, configure, host and administer the surveys enabled a reduction of time and cost. Further benefits of the researcher hosting and administering the survey software independently include better flexibility in the visual and structural presentation of the survey (e.g. changes can be made to the survey's source code), hosting options (can install identical survey on a back-up server) and retrieval and back-up of data (full control of —and access to database).

6.4.2. Survey Software

Survey software criteria were based on location, equipment, student access and, of course, instrument suitability. In this instance, PCs in the computer lab were configured to grant all individuals in the population access to the World-Wide-Web (WWW), while local network and system access was highly restricted. The likelihood of finding a free software solution with features suitable for such described environment was therefore considered to be highest in the web-based survey category. A survey made available online would also mean that participants unable to complete a submission on the day of the VCS software trial could do so at a later time, potentially from a different location. Indeed, a few participants did actually take advantage of this option.

Although Wright (2005) did not assess or even make mention of free survey software, one can only assume that these could potentially provide further advantages. Because of budget constraints, a number of free survey solutions were assessed. Some were considered on 'face value' (based on features listed on their respective websites), others more thoroughly by 'install and trial' in a local test-environment. The free and open source *LimeSurvey* (version 1.70+, build

4470) stood out as the most suitable software for this study, as it contained the features needed to mirror the paper-based structure of the instruments to be implemented (in particular the style and layout of some Likert-scales). It also provided export features to file formats desirable for subsequent data analyses, e.g. formats native to Microsoft Excel and the Statistical Package for the Social Sciences (SPSS). Technical requirements and dependencies of LimeSurvey, specifically the combination of PHP (Hypertext Pre-Processor) and MySQL, suited the researcher's prior knowledge and facilities provided by the server host.

Thorough evaluation of LimeSurvey was conducted in a local (off-line) server environment. Cumbersome menu navigation and survey creation were compensated by unparalleled features when compared to other free software trialled. Specifically, it offered the range of input modes and flexible layout options required by the survey items (e.g. plain text, Likert scales with multiple sub-items and yes/no input). An inbuilt template editor allowed for the manual editing of HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) code to deal with the rather unappealing and sometimes messy default survey templates (both visually and structurally). The researcher optimised parts of the HTML and CSS code and designed a new user interface template for use with the VCS surveys (as shown in Figure 6.10).

In the case of unforeseen problems with the primary host server, LimeSurvey was also installed on a secondary server as a backup. The inbuilt export feature eased the mirroring of created surveys to the back-up server, and a URL to this host was made available as an alternative location should the first server become unavailable. At the end of the week, the primary host had performed flawlessly and there was no need to employ the backup location.

A complete set of screen-shots of the final survey presentation and items are available in Appendices S & R.

6.5. Collected Data

As revealed by Tables 6.4, 6.5 and 6.6, the actual number of participants in the study ended up below the initial target (the entire population).

Contributing factors to these numbers may have been related to:

VCS :: Human Computer Interaction Evaluation

Progress: 0% 100%

2: Overall User Reactions

Please select the numbers which most appropriately reflect your impressions about using the Virtual Classroom Simulation. Not Applicable = NA.

Overall reactions to the Virtual Classroom Simulation:

	1	2	3	4	NA	
Terrible	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Wonderful
Frustrating	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Satisfying
Dull	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Stimulating
Difficult	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Easy
Inadequate Power	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Adequate Power
Rigid	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Flexible

<< prev next >>

[\[Exit and Clear Survey\]](#)

Figure 6.10.: Screenshot of a VCS HCI Survey Page

Time — particularly for the HCI Survey, which was the final item in the tutorial (a large number of students indicated that they did not have sufficient time to complete this item).

Incomplete and duplicate responses — some survey submissions were incomplete or submitted twice and, hence, not usable for data analysis. One explanation for incomplete and/or duplicate responses is that a few students who ran out of time resubmitted a (complete) survey in their own time (e.g. from home).

Student absence — e.g. due to sickness, other commitments or personal reasons.

The paper-based VCS STIC Questionnaire was handed out to students attending the lecture one week prior to the VCS introduction and collected by tutors in subsequent tutorial classes. A low number of responses could be attributed to students forgetting to answer the questions or returning the questionnaire.

Table 6.4.: VCS STIC Questionnaire Respondents and Gender Distribution

	Frequency	Percentage
Female	50 (of 117)	~42.7%
Male	34 (of 78)	~43.6%
Total	84(of 195)	~43.3%

Table 6.5.: VCS Education Survey Respondents and Gender Distribution

	Frequency	Percentage
Female	100 (of 117)	~85.4%
Male	59 (of 78)	~75.6%
Total	159 (of 195)	~81.5%

6.6. Qualitative Procedures Employed

A large amount of qualitative information was collected from the population through the STIC Questionnaire, VCS Education Survey and the VCS-related assignment. Section 7.2 in Chapter 7 describes two computer programs used to assist in survey data analysis and validation; SPSS Text Analysis for Surveys v2.1 (SPSS TAS) and RapidMiner v5.0.

6.6.1. Computerised Text Analysis (CTA)

SPSS TAS is a “linguistics-based solution specifically designed for categorizing or ‘coding’ survey text responses” (SPSS Inc., 2005, p. 2). In this study, SPSS TAS was primarily used to organise data, assist in theme identification, coding and categorisation of survey responses. The software also facilitated in the transformation of some data into a numerical form (quantification) and the subsequent export to formats supported by statistical analysis software.

Bright and O’Connor (2007) suggest that computerised text analysis (CTA) has the capacity to allow more data to be analysed within a shorter timeframe and improve rigour. Traditional text analysis (TTA), on the other hand, can be a tedious

Table 6.6.: VCS HCI Survey Respondents and Gender Distribution

	Frequency	Percentage
Female	58 (of 117)	~49.6%
Male	28 (of 78)	~35.9%
Total	86 (of 195)	~44.1%

task and more likely prone to human error (e.g. accuracy in frequency counts). In a study comparing TTA and CTA (SPSS TAS) performance, Bright and O'Connor (2007) found no noteworthy differences in their results. A suggested drawback of using CTA is that researchers are prone to losing 'closeness' to their data, which could happen as a result of text segmentation and loss of context (Bazeley, 2007; Weitzman, 1999). A counter-argument, which expresses the exact opposite, proposes that CTA brings researchers too close and thus causing them to be "bogged down in their own data" (Bazeley, 2007, p.8) or "drowning in detail" (Maclaran and Catterall, 2002, p. 37).

6.6.2. Objectivity and Truth

Some caution must, however, be exerted when adopting CTA for qualitative research. As Ary et al. (2009, p. 494) point out, "the computer cannot reflect or transform data into meaningful findings", thus interpretation and making sense of qualitative data is still in the hands of the analyst (Patton, 2002). Indeed, qualitative analysis is a subjective process:

For it is true that qualitative analysis, being inevitably subjective, cannot ascertain the accuracy and validity of its findings in the manner of an exact science. One and the same topic may invite different qualitative appraisals of almost equal plausibility; and no accumulation of evidence will determine, in an objective way, which is closer to truth. (Kracauer, 1952, p. 641)

Thus, in constructing concepts and categories, and analysing (coding) and categorising the data, the analyst exerts his or her own interpretation (Strauss and

Corbin, 1990). Although objective truth does not exist in qualitative analysis (Patton, 1987, p. 166; Kracauer, 1952, p. 641) efforts should still be made to minimise subjective bias. Shaw draws focus to Scriven's (1996; 1972) notion of 'goal-free evaluation':

'Goal-free evaluation' is Scriven's ideal means of reaching an evaluative judgment, by which he means that evaluators should proceed with a veil of ignorance regarding the goals held by programme managers and other stakeholders. Goals bias the evaluator. Rather than ask 'there are the objectives. Have they been achieved?', Scriven would prefer, 'Here is the programme. What are its effects?' Kept in ignorance of the programme goals, the evaluator will search for all outcomes. (Shaw, 1999, p. 23)

Patton (1987, p. 36) suggests that goal-free evaluation is particularly suited for qualitative methods as it may "eliminate the perceptual biases introduced into an evaluation by knowledge of goals" and "maintain evaluator objectivity and independence through goal-free conditions".

6.6.3. Bias Reduction and Validation

In this study, qualitative questions were designed to elicit relatively short statements. Although the instruments were comprised of a large number of questions, responses considered 'long' were typically only 5-6 sentences. The risk of 'getting lost' in the information was therefore not a major concern, but the analyst was nonetheless mindful about keeping the right 'distance' to the data.

While the features of SPSS TAS extends beyond that of basic word count (it also takes into consideration grammatical structure and meaning), human intervention and interpretation is still imperative (Stanford University, Social Science Data and Software, 2008). Some measures were therefore implemented to substantiate the validity of the findings. Where qualitative data were quantified for use in statistical analysis, no less than five individuals were employed to independently rate responses against the criteria guidelines. Inter-rater reliability was recorded and analysed to validate both the guidelines and detect potential bias exerted by the researcher.

Many of the survey questions aimed to draw out terms specific to the VCS and/or the educational context in which it operates. The data therefore lent itself well to

word frequency count as a means to establish categories most likely to be significant. Thus, a large portion of the researcher's subjective interpretation and categorisation (using SPSS TAS) was validated by cross-checking with reports generated by the open source data mining (DM) software RapidMiner. The software "allows the design of complex nested operator chains to solve a wide variety of learning problems effectively and efficiently" (Wu et al., 2008, p. 2), and Jungermann (2009) presents information to suggest that this is one of the best DM tools available. As will be detailed in Section 7.2.2 of Chapter 7, RapidMiner was thoroughly configured, using a number of operator chains, to extract the most significant tokens (words) from the dataset. The resulting RapidMiner ExampleSet (data table), which contained tokens and frequency counts, were compared to the categories established by the researcher.

This research study appreciates the value of mixed-methods research design. Triangulation and the use of exploratory, explanatory and confirmatory models are implemented to better understand the interactions between the variables under study. Some challenges in qualitative research were described in this section, as well as the measures put in place to alleviate these. Chapter 7 is entirely dedicated to the qualitative measures used in this study and the findings from the textual analysis. The remainder of this chapter is concerned with the statistical procedures employed for quantitative data analysis and instrument validation.

6.7. Statistical Procedures Employed

A number of statistical methods and procedures were employed in this research study for the purpose of data analysis, as well as instrument reliability and validity assessment. This section provides background information about these methods and procedures, while later chapters will show how they were implemented in the context of this study.

6.7.1. Scale Reliability: Cronbach's Alpha

Since all items in a scale typically aim to measure the same thing, they should be correlated with each other to maintain internal consistency (Bland and Altman,

1997). Cronbach's alpha (α) is a widely used index in *Classical Test Theory* (CTT) statistics for gauging a scale's reliability. This is achieved by determining the internal consistency or average correlation of the instrument items (Santos, 1999; Curtis, 2005). Test-retest or replication procedures using Cronbach's alpha may also be used in determining an instrument's stability (Cronbach, 1951). Test-retest involves multiple administrations of the same instrument for the same sample, while replication involves different samples (Hendrickson et al., 1993). A test-retest procedure was not feasible for this study, and replication could not be achieved since some items in the instruments used were altered. Where data were available, however, reliability indices from this study were compared to those from the original scales.

Cronbach α sits in a range from 0 to 1, where a higher number indicates better reliability for a scale. Exactly where the threshold value for good reliability sits can, however, be debated. Santos (1999) and Bland and Altman (1997) deem α at .70 as the acceptable cutoff value while others argue that one should generally aim for a reliability of at least .80 (Landis and Koch, 1977; Nunnally, 1978 in Bravo and Potvin, 1991).

A problem with CTT in educational assessment is that "the basic unit of measurement is not the person or the item, but rather the performance of the person relative to the item" (Keeves and Alagumalai, 1999, p. 24). Alagumalai and Curtis (2005, p. 10) point out that "CTT has limited effectiveness in educational measurement" and prescribe an alternative axiomatic measurement theory; the Rasch measurement model.

6.7.2. Confirmatory Factor Analysis: Instrument Reliability and Validity

A number of questions (items) employed in this research study's quantitative survey aim to measure aspects that are not easily observable. For example, in order to assess participants' attitudes towards the VCS it would be useful to obtain an insight into their attitudes, experience and abilities with using computers in general. These factors are important because they could impact on the users' perceived usability and usefulness of the VCS, and, hence, how they inform research problems of this

study.

Although no single question can elicit this information, it is possible to combine a set of question items (a *scale*) that, when combined, measures an overall *construct*. Here, the term ‘construct’ is defined by “a set of items and the responses to those items” (Wilson, 2005, p. 39). In this context, the construct is often referred to as a *latent variable* or *factor* because it cannot be observed directly (Byrne, 1989; Diamantopoulos and Siguaw, 2000). The questions/items of the construct on the other hand, which reflect the latent variable, are often referred to as *reflective indicators*, *manifest variables* or *observed variables* (Byrne, 1989; Diamantopoulos and Siguaw, 2000; Curtis, 2005). In other words, manifest variables (the measured scores) enable the indirect observation of the construct of interest (Byrne, 1989). This is exemplified in Figure 6.11, which illustrates a measurement model composed by a latent variable and a number of manifest variables.

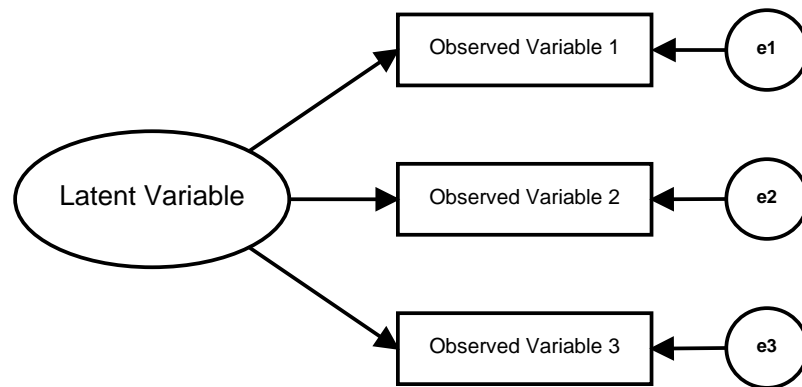


Figure 6.11.: Measurement Model

In order to learn more about the computer skills and attitudes construct, the researcher employed an adapted version of a well-established scale (a scale is sometimes referred to as an *index*, e.g. Adams and Wu (2002)) called “Comfort With and Perceived Ability to Use Computers”. Before the information elicited from this scale can be used in the analysis of data, however, it should first be ascertained that the observed variables actually measure the hypothesised relationship with one or more latent constructs. The most common statistical technique employed for this purpose is called *factor analysis* (Byrne, 1989). Two

basic types of factor analysis exist: *Exploratory Factor Analysis* (EFA) and *Confirmatory Factor Analysis* (CFA). EFA is typically performed when the underlying latent structure is not known (i.e. you explore for any factor(s) that may be underlying the manifest variables), while CFA is used to confirm or reject an already hypothesised *measurement model* (Byrne, 1989). The measurement model “describes how each latent variable is measured or operationalized by corresponding manifest indicators” (Diamantopoulos and Siguaw, 2000, p. 4).

With regard to the instruments used in this research study, the latent constructs were either explicitly or implicitly implied in related literature and/or in the scales themselves. CFA was therefore the preferred technique used for ascertaining the hypothesised models’ fit with the observed data. LInear Structural RELations (LISREL, Jöreskog and Sörbom, 2006) is a computer program designed to do *covariance structure analysis*; a multivariate statistical technique which includes (confirmatory) factor analysis (Diamantopoulos and Siguaw, 2000). In this research study, LISREL (version 8.80 for Microsoft Windows) was employed for carrying out CFA. Reports/outputs from the software, and a discussion of the statistical output, are presented in Chapters 8 and 9.

The aim of CFA is to determine if, and to what extent, the hypothesised model fits the observed data; the degree to which it is reflected by *factor loadings* and a number of *goodness-of-fit* statistics. ‘Factor loadings’ is a term used to indicate the relationship between the manifest variables and the latent variables they intend to measure (Byrne, 1989). One can therefore expect that a specific questionnaire item will load highly on the factor(s) it was designed to measure, and negligibly on other factors (Byrne, 1989). Shevlin and Miles (1998) propose three levels to indicate the degree of factor loadings; 0.3 as low, 0.5 as medium and 0.7 as high. According to Byrne, it is highly unlikely to achieve a perfect fit between a hypothesised model and the observed data, and the mis-fit (or differential) between the two is termed *residual*. Byrne (1989, p. 5) summarises the model-fitting process as follows:

$$\mathbf{Data} = \mathbf{Model} + \mathbf{Residual} \text{ (error)}$$

where

Data represents the observed measurements based on the sample.

Model represents the hypothesized structure underlying the observed variables.

Residuals represents the difference between the hypothesized model and observed data.

6.7.2.1. Assessment of Model Fit

A model's fit refers to the consistency between observed data and the hypothesised model. The LISREL program generates a number of indices (labelled as 'Goodness of Fit Statistics') that may be useful for assessing this fit (or misfit). Diamantopoulos and Siguaw (2000) advise that not all of these indices are needed in order to make an informed decision concerning the model's overall fit. "For practical purposes", they say, "the results of the chi-square test used in conjunction with the RMSEA, ECVI, standardized RMR, GFI and CFI indices should be more than sufficient" (2000, p. 88). In this study, where assessment of model fit is conducted, the indices that will be examined and reported on are as follows:

Chi-square statistic (χ^2), denoted by LISREL as 'Minimum Fit Function Chi-Square', "is the traditional measure for evaluating overall model fit in covariance structure models" (Diamantopoulos and Siguaw, 2000, p. 83). A statistically significant χ^2 (large χ^2 value) implies an imperfect fit and a possible rejection of the model (Jaccard and Wan, 1996 in Diamantopoulos and Siguaw, 2000). The degrees of freedom (df) serve as a standard by which to judge whether χ^2 is large or small. A good fit is implied where $\chi^2/\text{df} \leq 3.0$, and a p-value $\geq .05$ is also recommended (Segars and Grover, 1993). Chi-square should, however, be used with caution because it is sensitive to sample size (Anderson and Gerbing, 1984; Bollen and Long, 1993), departures from multivariate normality and it assumes that the model fits perfectly in the population (Diamantopoulos and Siguaw, 2000); using chi-square in conjunction with other indices is therefore recommended .

Root mean square error of approximation The RMSEA is generally regarded as one of the most informative fit indices. It shows how well the model, with unknown but optimally chosen parameter values, would fit the population covariance matrix if it were available. Values less than 0.05 are indicative of good fit, between 0.05 and 0.10 of mediocre fit and >0.10 of poor fit (Diamantopoulos and Siguaw, 2000).

Root Mean Square Residual Standardised RMR is a summary measure of standardised residuals. Values indicative of a good fit should be below 0.05 (Diamantopoulos and Siguaw, 2000).

Goodness of Fit Index The GFI is an absolute fit index that indicates the “relevant amount of variances and covariances accounted for by the model and thus shows how closely the model comes to perfectly reproducing the observed covariance matrix” (Diamantopoulos and Siguaw, 2000, p. 87). A GFI value $>.90$ usually reflects an acceptable fit (Diamantopoulos and Siguaw, 2000; Shevlin and Miles, 1998).

Adjusted Goodness-of-Fit Index The AGFI is the GFI adjusted for degrees of freedom in the model. Similar to the GFI, values $>.90$ usually reflect an acceptable fit (Diamantopoulos and Siguaw, 2000).

Parsimony Goodness-of-Fit Index The PGFI takes into account model complexity, with acceptable values much lower than the GFI and AGFI (Diamantopoulos and Siguaw, 2000). Byrne (1998) informs that it should not be unexpected to see PGFI indices in the range of $.50$, where χ^2 statistics are non-significant and goodness-of-fit indices are in the range of $>.90$.

Comparative Fit Index CFI values should range between 0-1, with values close to 1 representing a good fit (Diamantopoulos and Siguaw, 2000).

“Fit indices”, Bollen and Long (1993, p. 157) say, “should not be regarded as measures of usefulness of a model”. They argue that the selection of a model should be a subjective process that involves the use of judgment. While the above indices may assist in making an informed decision concerning the model’s overall fit, this

research study will also employ other approaches for scale analysis. Final model assessment will then be made when all of these processes have concluded.

6.7.3. Rasch Measurement Model

Andrich (2004) highlights some controversy regarding the Rasch model's belonging to a class of measurement approaches known as the latent trait test theory, or more commonly the *item response theory* (IRT). This controversy, Andrich (2004) explains, is mainly due to conflicting perspectives about the differences between the two paradigms. When making the choice between the Rasch and IRT models for use with dichotomous items, Andrich (2004) points out that the former requires the data to fit the model while the latter requires that the model fit the data. Comprehensive developments on the original Rasch model have resulted in a number of derivative measurement models often referred to as the Rasch *family of models*. Keeves and Masters (1999) list a number of advantages of Rasch measurement, including:

- the ease of equating tests on the same scale (and testing whether two instruments measure the same dimension),
- sample independence and ability to detect item bias (differential item functioning),
- identification of inconsistent behaviour in test-takers,
- subset of calibrated items may be used to estimate test-takers' performance to a specified degree of accuracy;
 - computerised extension may adaptively pick items close to a person's level of performance, whereby shortening a test by eliminating items that are too easy, difficult or non-discriminating.

Unlike CTT, a scale is independent of both the test items and the sample of persons employed in calibration when using Rasch scaling (Keeves and Masters, 1999). Rasch measurement introduces probabilities of item response, which considers the odds of a person giving a correct or favourable response. If a person's ability equals that of an item's level of difficulty, the probability of the

person responding correctly to that item is commonly 50% (Keeves and Alagumalai, 1999). Items must, as such, be calibrated in order to determine their difficulty; a process that may involve the rejection of items and/or persons (Keeves and Alagumalai, 1999).

A *measure*, Vernon (1978, p. 3) says, is “the numerical expression of a certain standing on, or amount of, a variable”. But as Keeves and Masters (1999) point out, characteristics of interest must sometimes be measured indirectly (e.g. latent traits). This is a cardinal idea of IRT, as it assumes the presence of an underlying factor that is being measured (which is also true for CTT in the concept of ‘true score’) (Keeves and Alagumalai, 1999). *Measurement*, then, implies that only one characteristic is quantified at a time (Lumsden, 1957 in Alagumalai and Keeves, 1996). Because it has been argued that more than one latent trait is involved in the measurement of achievement and attitudes, the construct of *unidimensionality* was introduced (Wright and Stone, 1979):

A unidimensional test may be defined simply as a test in which all items are measuring the same thing. (Alagumalai and Keeves, 1996, p. 1)

Unidimensionality suggests that there might be more than just one latent process involved when responding to a set of items, but unidimensionality will hold as long as each item is affected by the same processes in the same form (Bejar, 1983, in Keeves and Alagumalai, 1999). Thus, before a total test score is calculated (CTT) or a scaled score is estimated (IRT), the condition of unidimensionality must hold (Keeves and Alagumalai, 1999).

It would seem desirable that as a first step in the construction of a scale of measurement confirmatory factor analysis should be employed to examine the covariation among responses to a set of test or attitude scale items. (Keeves and Alagumalai, 1999, p. 31)

Accordingly, CFA was employed in this study to establish the presence of a single latent trait and to determine if unidimensionality would hold. The Rasch measurement model was subsequently implemented in order to ascertain the data’s fit with the model.

6.7.3.1. Rating Scale Model

When measuring cognitive abilities and skills, it is common to score test items dichotomously as incorrect or correct (zero/one) and the original ('simple') Rasch model is suitable to determine item difficulty and person ability in these cases. As Masters (1982) points out, however, there are *observation formats* other than the dichotomously-scored responses for which the general Rasch model was intended. Formats that record ordered levels of response include repeated trials, counts, partial credit and rating scales (Masters, 1982).

Some tests award *partial points* depending on the degree of correctness or 'partial knowledge' (Athanasou and Lamprianou, 2002). The *Partial Credit Model* (PCM), which belongs to the family of Rasch models, is suitable for the analysis of polychotomously scored data where partial points are awarded to each item (Masters, 1999). The PCM works on ordered levels of performance, where "response alternatives are free to vary in number and structure from item to item" (Masters, 1982, p. 150).

The *Rating Scale Model* (RSM), on the other hand, is useful in the analysis of polychotomous items for which the response alternatives are fixed (e.g. "strongly agree", "agree", "disagree") (Andrich, 1978; Masters, 1999). The RSM is not typically concerned with the cognitive skills and abilities of the test-taker, but rather focus on other traits such as attitudes and personalities (e.g. questionnaires/surveys).

The RSM is particularly useful in this study as the nature of the scales employed is mostly polychotomous and use fixed response categories. Since these questionnaires do not measure cognitive qualities, the terms 'person ability' and 'item difficulty' are therefore replaced with 'attitude' and 'affective value' respectively (as suggested by Müller, 1987).

6.7.3.2. Model Fit Statistics

The Infit Mean Square (Infit MNSQ) is one of the most important statistics in determining the quality of measurement in the Rasch model (Athanasou and Lamprianou, 2002). It is used to measure the mismatch (residuals, discrepancy) between predicted responses by the Rasch model and the actual responses. A large

Infit MNSQ means that the item is ‘misbehaving’ and possibly violating the unidimensionality (e.g. not measuring the same thing as the other items) (Athanasou and Lamprianou, 2002).

In the context of Rasch models, Infit MNSQ statistics have an expectation of 1.0 and can range from 0 to infinity (Wright and Linacre, 1994). While there are no hard rules on cut-off scores, Wright and Linacre (1994) suggest an Infit MNSQ range of 0.6 — 1.4 as reasonable for a rating scale (survey). However, as Athanasou and Lamprianou (2002) point out, this range is not always appropriate and they recommend that it is applied in a judicious manner. Adams and Wright (1994) argue that data can never be perfectly compatible with any measurement model; given a sufficiently large number of observations, there will always be an incompatibility between any data and any model (Martin-Löf, 1974; Gustafsson, 1980 in Adams and Wright, 1994).

Rasch measurement was undertaken using ‘ConQuest’; a generalised item response modelling software developed by the Australian Council for Educational Research (ACER). A description of how RSM was accomplished for the relevant scales in this study, as well as reports on the model fit statistics, are included in Chapters 8 and 9.

6.7.4. Path Analysis

Structural Equation Modelling (SEM), and its corresponding application in regression path modelling, highlight that “data analysis is an interaction between theory and data where the conclusions drawn are dependent on the starting theoretical/conceptual model and the path chosen to make modifications to the model” (Kühnel, 2001, p. 91). Kühnel further iterates the role of exploration of a model, an attempt to explain the relationships within the model, and importantly the confirmation of the model in confirming the viability and functionality of the advanced theoretical model. In undertaking any quantitative data analysis, there are a number of assumptions the researcher takes and there are associated errors and risk of arriving at wrong conclusions. Kano (2001) and Kühnel (2001) argue that all priori assumptions must be guided by theoretical arguments. Thus, any “substantive theory in research helps to decide which data are necessary, how data

can be sampled, which statistical model seems to be appropriate, and how the results of data analysis contributed to knowledge” (Kühnel, 2001, p. 93). Figure 6.12 highlights this paradigm applicable for all research in education and the social sciences.

NOTE:
This figure is included on page 169 of the print copy of
the thesis held in the University of Adelaide Library.

Figure 6.12.: Statistical Data Analysis as Interaction Between Theory and Reality
(Kühnel, 2001, p. 95)

Thus, empirical data analysis is grounded in methodological and statistical concepts. Kühnel (2001, p. 96) contends the process associated with structural equation modelling adds real value to experiences of the real work and abstract ideas.

“The notion of causation has been one of the most dominant analytic concerns in the history of science” (Olobatuyi, 2006, p. 2). Causation, as highlighted by Kühnel (2001) and Olobatuyi (2006), has also played a central role in the construction of research design in this study.

The theoretical model of the current study was introduced in Chapter 1. The model was built around scales from the VCS HCI Survey and the VCS Education Survey. While these surveys were briefly introduced earlier in this chapter, the discussion has yet to elaborate on their scales and associated items. This will, however, be addressed in Chapters 7, 8 and 9. These chapters will also, where applicable, report how the statistical procedures described earlier in this chapter were utilised to test the reliability and validity of a number of scales in the surveys.

These scales can be observed in Figure 6.13, which provides a more detailed view of the theoretical model advanced in Chapter 1.

The discussion earlier pointed out that scale structure—and item analyses are employed to ascertain how well hypothesised models fit the observed data (CFA) and how well items fit the Rasch model. Thus, these techniques are used to ascertain the reliability, validity, bandwidth, fidelity and separation index of a scale and its items. Path Analysis (PA), on the other hand, is a statistical approach for measuring the linkages between variables (Asher, 1988). Schumacker and Lomax (2004) caution the use of the term ‘causal modelling’ because PA does not discover causes, but *tests theoretical relationships*. This re-affirms Kühnel’s (2001) hypothesis between the relationship between ‘reality’, ‘data’, and ‘theory/model’. Knowledge about the degree, or magnitude, of linkages between variables may therefore be used to inform on variables’ causal relationships (Asher, 1988). In the context of this study for example, PA will be used to investigate the relationship between a student’s overall reactions to the VCS (VCSREA) and their attitudes towards computers (COMATT). The examination of causal relationships between two or more variables would not, however, be meaningful if the integrity of those variables had not been established. Thus, if one or more items belonging to the construct of COMATT did not actually fit the model (did not measure the construct), the causal relationship between COMATT and VCSREA would not be accurate. PA should therefore be performed only after the validity and reliability of all variables involved have been established. As will be demonstrated in this study, this may involve the rejection of items. Chapter 10 will present how rejected items impose adjustments to the theoretical model and, thus, the PA.

6.8. Summary

This chapter provided details about the study’s research design. It follows a strategy which engaged a carefully chosen population for which the VCS was developed. The triangulation mixed methods employed in the research design involved both concurrent and separate collection of quantitative and qualitative data from the population. Pending structure analysis of the involved scales, information that is confirmed to be valid and reliable will be merged to better

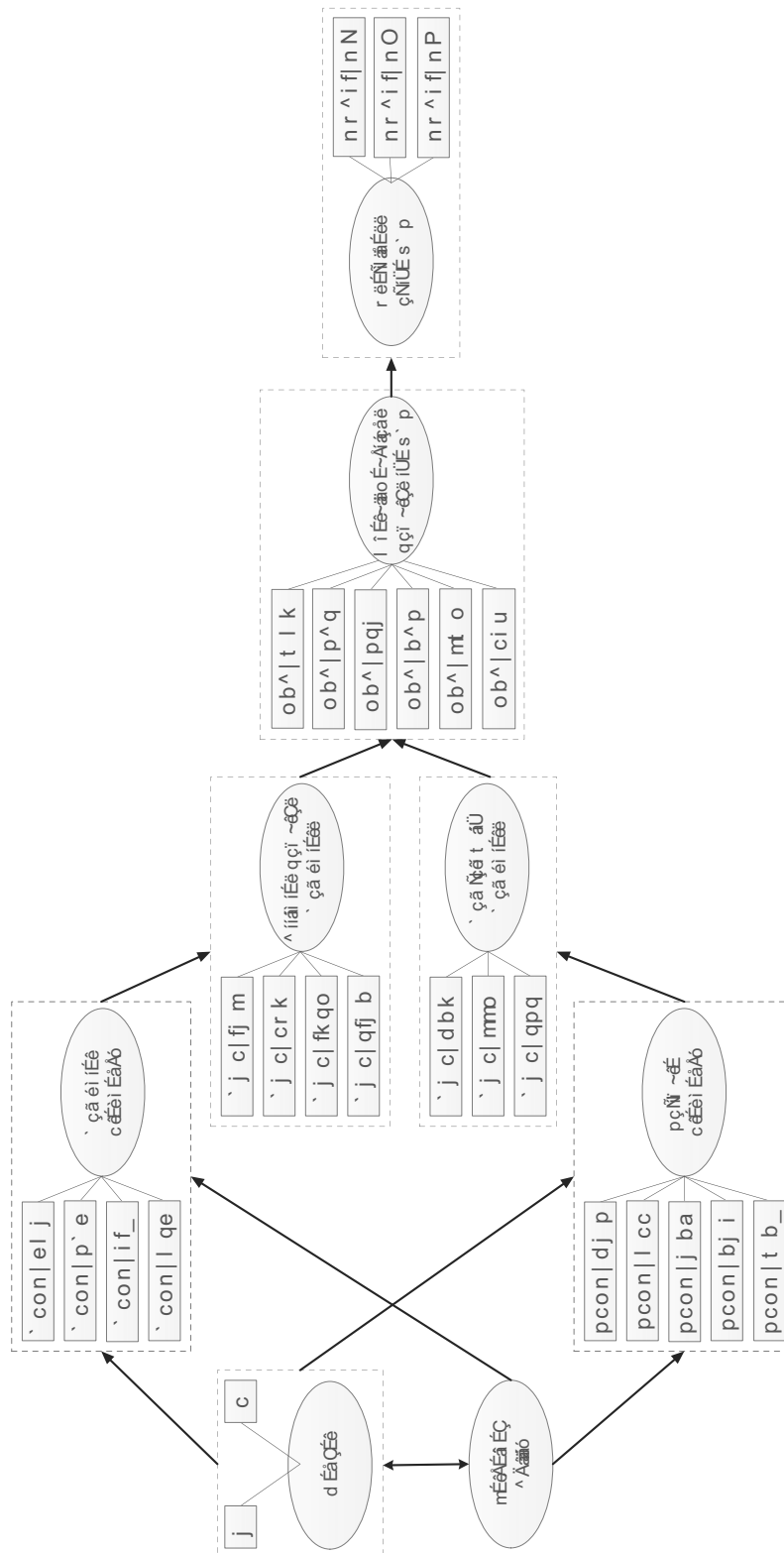


Figure 6.13.: Detailed View of the Theoretical Model

Chapter 6. Research Design

understand the research problems of this study (Creswell, 2008, p. 557). This entails the statistical procedures introduced in this chapter, which described essential techniques required to ascertain the reliability and validity of scales and items.

Chapter 10 will demonstrate how path analysis was employed to examine the refined theoretical model and determine causal relationships between a number of variables pertinent to this study. Findings from this analysis will be used in conjunction with other data to address the research problems and produce concluding observations about the VCS.

Chapter 7.

VCS Qualitative Measures

Thus the apparent simplicity of qualitative “data” masks a good deal of complexity, requiring plenty of care and self-awareness on the part of the researcher.

(MILES AND HUBERMAN, 1994, P. 10)

Research Problem 3 (RP3) seeks to investigate the VCS' capacity to advance education students' mental models about teaching and learning and assist in remediating misconceptions that may be present.

Problem 3. To what extent is the VCS useful in reshaping preservice teachers' mental model about teaching and learning, as well as advancing their pedagogical content knowledge (PCK)?

- How can this be accurately measured?
- How can the impact of the VCS be further improved?

Several aspects of this problem can be addressed only through the acquisition of information directly from the target population. Access to the population was, however, restricted to one lecture (50 minutes) and one tutorial session (50 minutes) per group. Hence, the strategy for collecting information required careful consideration. Since the problem seeks to establish a progressive change of education students' mental models as a result from using the VCS, the research

Chapter 7. VCS Qualitative Measures

had to be undertaken over a relatively short period of time. A number of qualitative measures were constructed and validated to collect the necessary data.

Three instruments workable within the context of the education course “Student-Teacher Interaction in the Classroom 1” (STIC 1) were devised for capturing qualitative information from the students:

1. STIC Questionnaire (described in Section 6.3.1, Chapter 6)
2. VCS Education Survey (described in Section 6.3.4, Chapter 6)
3. STIC 1 Assignment (described in Section 6.3.6, Chapter 6)

Figure 7.1 provides a reminder of the timeframe and activities in which the population was involved with the VCS.

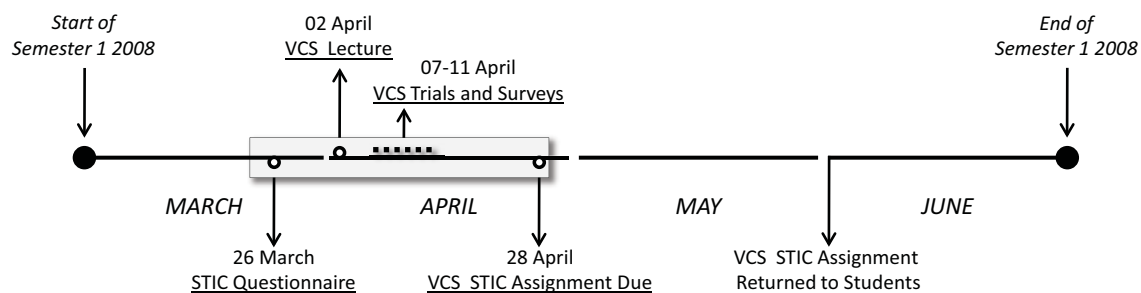


Figure 7.1.: Timeframe of the Population’s Involvement with the VCS

7.1. Data

The STIC Questionnaire, which will be discussed in more detail in Section 7.3, was distributed by the course coordinator to all students attending the STIC class in the week prior to the VCS Introductory Lecture. 84 students completed and returned the questionnaire and these responses were transcribed into the Microsoft Excel format.

As discussed in Chapter 6, Section 6.4, the VCS Education Survey was administered online. It received a relatively large response rate with a total of 159

students participating in the survey. Although all questions were mandatory and enforced by the survey software, a small number of respondents circumvented one or more questions by adding a short 'NA' (not applicable) text or similar in their answer. All information was captured directly into an electronic format and collected data were exported from the LimeSurvey software into the Microsoft Excel format.

All STIC 1 assignments were scanned into the PDF format after they had been marked. Student marks were included in the statistical analysis, while some of the submissions were jointly used in the qualitative analysis.

The next section describes the qualitative data analysis software employed to assist in organising and coding collected data.

7.2. Qualitative Data Analysis Software

Two main computer applications were used for qualitative analysis and mining in this research study: SPSS Text Analysis for Surveys v2.1 (SPSS TAS) and RapidMiner v5.0. These qualitative data analysis software can be useful when organising, searching and coding data. They do not analyse the data for you, however, but rather provide facilities to assist in the analysis (Creswell, 2008, p. 248). Thus, findings are generated by both software and researcher intervention, resulting in a mix of both software and researcher interpretations. This is not a characteristic unique to this study, however, as all social science research is based on assumptions (Morgan and Smircich, 1980) and is arguably subjective (Creswell, 2008; Marshall and Rossman, 2006). The findings and related discussions should therefore be regarded as estimates only. All narrative inclusions (quotes) are, however, exact representations from the respondents.

7.2.1. SPSS Text Analysis for Surveys

SPSS TAS was used in this research to categorise key concepts in the survey data. The software provides a number of useful features that ease this process, including automatic extraction and categorisation of terms and patterns in the qualitative dataset. It also provides a drag-and-drop functionality for manual fine-tuning of the

results. As noted in the SPSS TAS help document (accessible in the menu **Help** > **Text Analysis for Surveys...**), “text analysis is most powerful when performed in an iterative manner (extract, review, refine, reextract)”. Thus, as noted earlier, subjectiveness is an implicit element in this process. The researcher was, however, guided by literature around specific questions and this will be discussed in the sections to follow.

7.2.2. RapidMiner

While SPSS TAS is very useful in the extract-review-refine-reextract process, categories are produced by software and human interaction. SPSS TAS does not readily provide a feature for advanced token (word) filtering (e.g. length, stemming and stopwords), subsequent frequency count and reporting of the resulting dataset. The sophisticated software system RapidMiner v5.0 was therefore employed in some instances to further mine the qualitative data with the aim to establish the reliability of findings from using SPSS TAS. Collectively, SPSS TAS and RapidMiner complement each other and add further detail to the discussion.

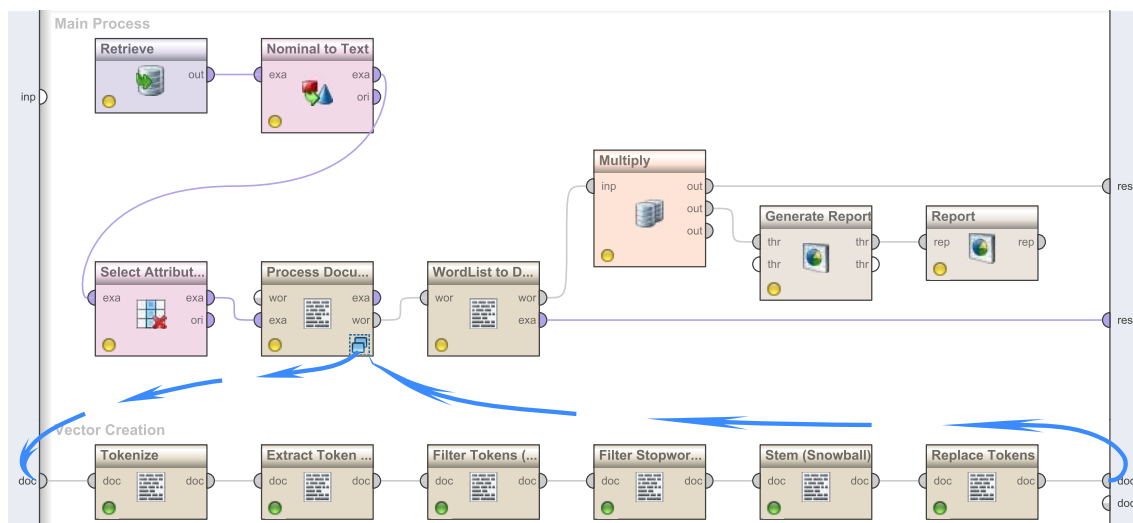


Figure 7.2.: RapidMiner Process Template for Analysis of VCS Qualitative Data

Figure 7.2 illustrates the RapidMiner operators (sequence of processing steps) used to filter, extract and report quantitative findings from the qualitative survey data. The operators perform the following tasks in sequence:

1. **Retrieve:** Reads the Excel file in which the qualitative data is held.
2. **Nominal to Text:** Pre-processing operator that transforms the data to plain text.
3. **Select Attributes:** Allows for the selection of one or more attributes (questions) to be processed. Question (attribute) responses are referred to as ‘documents’ in RapidMiner, while rows in the dataset are called ‘examples’.
4. **Process Documents From Data:** The selected attribute’s (question) documents (responses) are processed here through a number of sub-processes (‘Vector Creation’) in the following sequence:
 - a) **Tokenize:** Splits each response text into separate words (tokens). The algorithm employed to achieve this treats non-letters as word separators.
 - b) **Extract Tokens:** Extracts the number of tokens and adds it to the document’s metadata.
 - c) **Filter Tokens (by Length):** Dismisses words below or above a pre-defined length. As a general rule in this project, all words with less than 4 characters were removed from the dataset. Exceptions were made for responses where shorter words, such as ‘yes’ and ‘no’, were significant to the question (e.g. question 12; “Is self-efficacy affected by question answering?”).
 - d) **Filter Stopwords (English):** “A stop word may be identified as a word that has the same likelihood of occurring in those documents not relevant to a query as in those documents relevant to the query” (Wilbur and Sirotkin, 1992, p. 45), e.g. ‘the’, ‘and’, ‘but’, ‘with’, etc. This operator filters (removes) English stopwords from responses using RapidMiner’s built-in stopword list. The operator was disabled in the processing of questions where one or more stopwords (e.g. ‘no’) were relevant to the response analysis.
 - e) **Stem (Snowball):** The final operator used in the document processing was stemming using the Snowball algorithm. Kettunen et al. (2005, p. 476) describes stemming as “many-to-one mapping where semantically

related distinct word forms are reduced to identical stems”. The RapidMiner Stem operator reduces tokens to their root form, allowing words that otherwise would have been counted separately to be combined into one (e.g. ‘distract’, ‘distracted’, ‘distractibility’ are merged into a single word/attribute called ‘distract’). The Snowball algorithm does not use a dictionary/thesaurus and is therefore unable to take into account synonyms (e.g. it does not merge the words ‘response’ and ‘answer’).

5. **WordList to Data:** The Process Documents From Data operator provides a word list as an output. The WordList to Data operator converts this word list into a data set with a row for each word and associated word count (total occurrences and number of occurrences in each response). This dataset is useful for further processing, either in RapidMiner or exported to Excel.
6. **Generate Report:** Generates and exports a new report in either PDF, Excel, HTML or RTF format. Used in this study to provide an Excel file for post-processing.
7. **Report:** Adds report items to a report generated by ‘Generate Report’ operator.

The heuristics adopted by RapidMiner 5.0 parallels the qualitative data-analyses techniques advanced by Creswell (2008), Shank (2005) and Strauss and Corbin (1990). However, RapidMiner does not take meaning into account when words are counted. Thus, the word ‘knowledge’ may not always refer to the student entity’s knowledge (e.g. “to the best of my knowledge...”). Some discrepancies should therefore be expected between the qualitative analysis and the objective reporting generated by RapidMiner.

7.3. VCS STIC Questionnaire

The VCS STIC Questionnaire, which can be found in Appendix J, sought to extract and identify keywords of what the education students found most significant with regard to the teacher, student and question in the classroom interaction context. No

reference was made to the VCS, as the education students were to remain unaware of its existence when responding to the questionnaire. In partly answering the RP3 of knowledge into PCK for education students, the following specific questions were posed:

1. *Briefly identify at least three characteristics of a teacher that would be useful in the interaction in the classroom?*
2. *What factors affect student learning in the classroom?*
3. *In posing a question or task to a student, what do you bear in mind about the question/task?*

Participants were asked to include at least three factors/characteristics for each question. The majority of responses conformed to the format of the questionnaire (e.g. short, concise answers). RapidMiner was therefore particularly suitable for summarising the data using the techniques described in Section 7.2.2. Where a single response included the same word two or more times, it was counted only once. Words with a similar meaning were merged, as indicated in the tabular reports.

7.3.1. Question 1

Briefly identify at least three characteristics of a teacher that would be useful in the interaction in the classroom?

It is evident that skills/knowledge, enthusiasm and personal disposition are three important characteristics of a teacher that enhances interactions in the classroom. Table 7.1 provides the most frequently used words in the responses. The VCS software does not explicitly model the teacher entity, though some of these characteristics may be exercised in the simulation.

7.3.2. Question 2

What factors affect student learning in the classroom?

As shown in Table 7.2, the teacher, environment and subject matter were the most frequently mentioned factors that affect student learning in the classroom. A

Table 7.1.: Word Frequency, Question 1

Word	Responses
Skills/Ability/Knowledge/Intelligence	43
Enthusiastic/Passionate/Engaged/Encouraging/Positive	40
Approachable/Friendly/Interpersonal	33
Caring/Understanding/Compassionate/Empathetic	27
Patient	21
Confident	19
Clear Communication	19
Organised/Prepared	17
Listen	16
Voice	13
Humour	10

N=84

number of the factors listed in the table are also considered in the VCS. The teacher (the user), task, emotional comfort (mood), student knowledge (prior and current knowledge state), interest (as generated by question novelty), behaviour (distractibility) and interaction are all explicitly included in the software.

7.3.3. Question 3

In posing a question or task to a student, what do you bear in mind about the question/task?

An important aspect of the VCS is to emphasise the interaction between student knowledge and question difficulty. As surfaced in Table 7.3, these seem to be the most important factors to consider when preparing a question/task to a student. It could be argued, though, that student knowledge and question difficulty represent the same thing; consideration of student knowledge inherently involves question difficulty and vice versa.

The VCS also acknowledges the objectives/outcome of a task (e.g. factors in the results charts), if the question is interesting (novelty) and the importance of time (time penalty, student wait-time and time to ask a student). Since the VCS is subject/topic neutral, question clarity and relevance are not considered.

Table 7.2.: Word Frequency, Question 2

Word	Number of Responses
Teacher	54
Environment/Atmosphere	44
Topic/Content/Activity/Task	38
Physical & Emotional Comfort	16
Ability/Knowledge	16
Interest	14
Motivation	14
Behaviour	12
Engaging	12
Interaction	11
Background	8
Culture	8
Group	8

N=84

In the analysis of this question, it was found that the nature of responses might not be as suitable for raw word count as for those to the previous two questions. The way in which many responses were worded caused reason for concern that too much detail would be lost from an objective word count with RapidMiner. SPSS TAS was therefore employed to investigate the data in more depth, with more careful (manual) analysis of themes and categorisation of responses. As shown in Table 7.4, however, the two methods produced quite similar output. The most noticeable difference can be found in the emphasis of *question clarity*, a category into which the researcher (with the aid of SPSS TAS) fitted twice the responses than was the case with raw word count.

7.3.4. Summary

The VCS STIC Questionnaire provided some insight into the population's PCK. It highlighted what entities and attributes they ranked as having most value in the various teaching and learning contexts. The questionnaire was administered prior to the population's knowledge of the VCS. The next section interrogates data collected from the education students after the VCS trials.

Table 7.3.: Word Frequency, Question 3

Word	Number of Responses
Student Knowledge Understanding Capabilities	48
Question Difficulty Appropriateness	41
Question Clarity	22
Question Relevance	19
Objectives Outcome	18
Question Interesting Engaging	18
Time	6

N=84

Table 7.4.: Response Themes and Categorisation, Question 3

Theme/Category	Number of Responses
Student Knowledge Capabilities	50
Question Difficulty Level Appropriateness	45
Question Clarity	44
Question Interesting Engaging	23
Objectives Outcome	22
Question Relevance	20
Time	10

N=84

7.4. VCS Education Survey

The VCS Education Survey (Appendix R) sought to elicit qualitative information for use in further analysis. Table 7.5 lists the eighteen questions included in the survey in sequential order.

The question items may be placed in subgroups that address areas of specific interest to this study, e.g.:

- the usefulness of the VCS for teaching and learning,
- important characteristics of the student, teacher and task (entities),
- strengths and weaknesses of the VCS,
- theoretical concepts specific to STIC and the VCS,

Table 7.5.: VCS Education Survey: Question Items

Question	Question Text
1*	How do you view the educational value of the VCS?
2*	Do you believe the VCS is an effective means of training? How and Why?
3*	Do you believe the VCS generates a valuable learning experience? How and Why?
4	On which element/aspect of the simulation did you place most emphasis? Please comment.
5	From your understanding of the VCS, what are the attributes of the students that the teacher need to take into consideration when posing a task or question?
6	Can you list the attributes of a teacher, as presented either explicitly or otherwise in the VCS?
7	In posing a question or task to the class or a student, what does the teacher need to know to minimise distractibility and optimise learning?
8	What are the strengths of the VCS? Please give specific examples.
9	From your knowledge and experience in classrooms, does the simulation meaningfully resemble the real environment? Please comment.
10	What improvements need to be incorporated into the VCS as a teaching and/or training tool?
11	Explain how knowing about [the student's prior knowledge about the topic] is important for the teacher?
12	Is self-efficacy affected by question answering?
13	How could students' level of prior knowledge about a lesson topic be related to their level of distractibility?
14	Explain how consideration of the level of difficulty of a question is important for a teacher.
15	Annelie had a high level of distractibility after about 1 min into the lesson. Why do you think her level of distractibility increased and stayed quite high for the rest of the lesson?
16	What is Kennet like as a student?
17	If Maja's face showed less sadness during the lesson, why would this be?
18	From your understanding of the VCS, in what scenario(s) is a student's Knowledge State affected?

* Item responses were quantified and rater-reliability checked.

- knowledge and understanding of VCS students.

Figure 7.3 provides a visual representation of the above grouping.

7.4.1. Questions 1-3: Quantitative Transformation and Analysis

The VCS Education Survey consisted of eighteen open-ended questions. It was, however, useful to combine responses from a few of these questions with the quantitative dataset from the VCS HCI Survey (Chapters 8 and 9). Of particular interest are data concerning the educational usefulness of the VCS, as captured in the first three questions of the Education Survey. Information of this nature was not elicited from the VCS HCI Survey, which was more concerned with the users' attitudes towards computer technology in general and their satisfaction of the technical aspects of the VCS software (e.g. user-friendliness). A path analysis of a) the users' attitudes towards computers (Computer Familiarity Scale, Chapter 8), b) satisfaction with the VCS (User Interaction Satisfaction Scale, Chapter 9) and c) perception of the VCS' usefulness for teaching and learning (VCS Education Survey) allows for the inference of how each property may or may not affect the others and in ascertaining causality (Chapter 10). For this to be achieved, the qualitative data needed to be simplified or abstracted (e.g. data reduction) into numbers (Miles and Huberman, 1994). Responses to question items 1, 2 and 3 were therefore transcoded into a quantitative representation for use in the conjunct analysis with other quantitative data:

1. Do you believe the VCS generates a valuable learning experience? (How and why?)
2. Do you believe the VCS is an effective means of training? (How and why?)
3. How do you view the educational value of the VCS?

The data reduction involved a sequence of steps necessary to ensure the integrity of the resulting information. These steps are outlined below:

1. Explore the general sense of the data (preliminary labeling and coding),

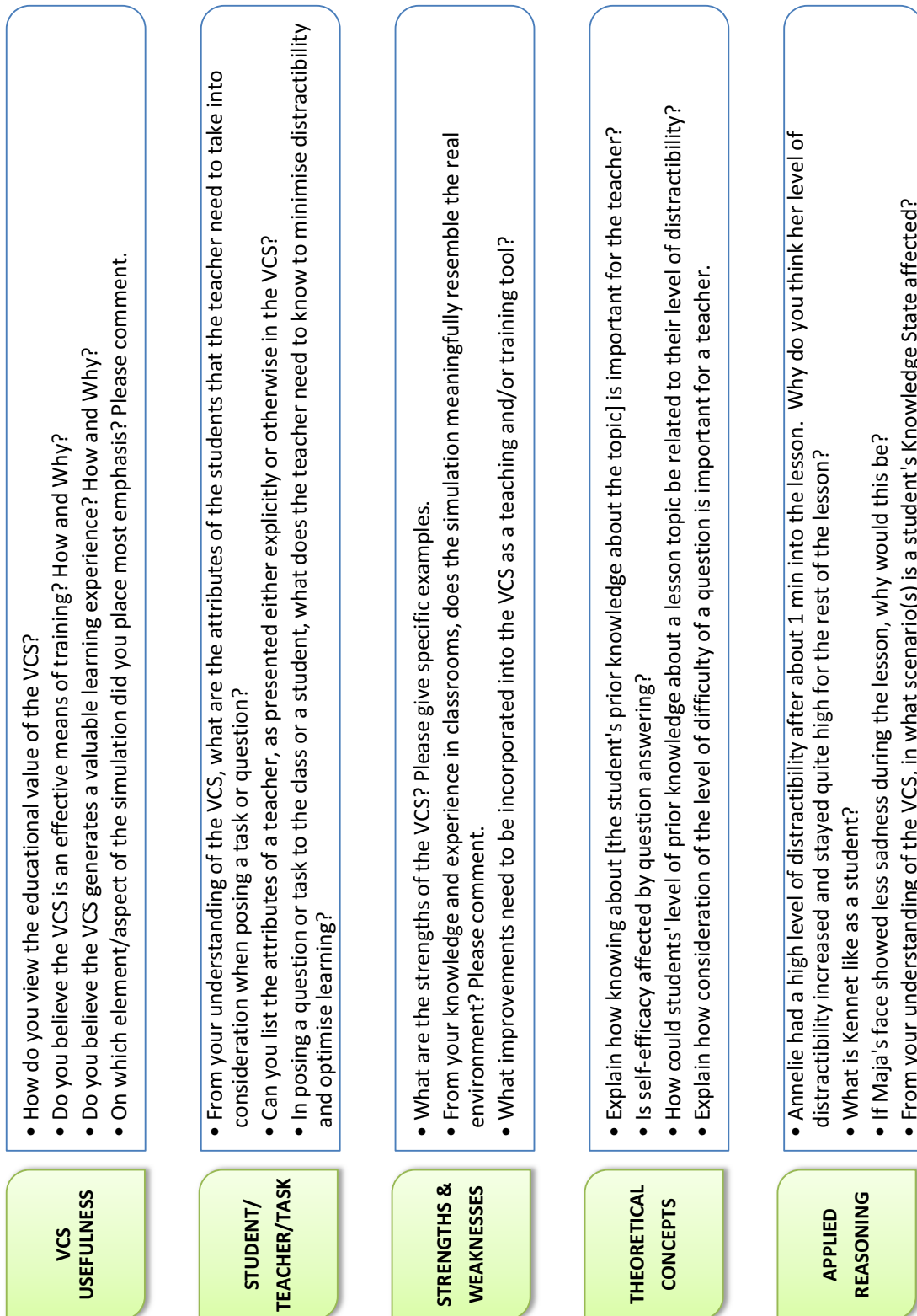


Figure 7.3.: Targeted areas of the VCS Education Survey

Chapter 7. VCS Qualitative Measures

2. Establish criteria guidelines, or category schemes (Castorr et al., 1990), for categorisation of responses,
3. Code and identify themes and response categories from the item responses,
4. Researcher to perform categorisation of responses, according to identified themes and criteria guidelines,
5. Employ objective raters to categorise responses in a similar manner,
6. Evaluate rater agreement.

The preliminary activities of coding, theme identification and response categorisation follow similar processes to those suggested by Creswell (2008) and Strauss and Corbin (1990) for analysing and interpreting qualitative data. Independent raters were employed to replicate the categorisation process in order to detect and moderate any subjectiveness imposed by the researcher.

7.4.1.1. Categorisation

The formation of response themes (Creswell, 2008), or what Strauss and Corbin (1990) refer to as categories or dimensions, was first and foremost based on the question design. All three questions were carefully developed to ease the transformation of responses from a qualitative to a quantitative form (e.g. the grading nature of question 1 and the 'yes/no' nature of questions 2 & 3). Established themes and their corresponding category schemes (Tables 7.6 and 7.7) were useful in the categorisation of item responses.

It was important that the categories and their criteria guidelines were sufficiently unequivocal to allow for a number of raters to be consistent in the categorisation of item responses.

7.4.1.2. Inter-Rater Agreement

In the process of data-reduction (quantification) the rater may be a source of measurement error (Landis and Koch, 1977). The researcher of this study may, for example, consciously or subconsciously categorise responses in a way that reflects

Table 7.6.: Categories and Schemes Established for Question 1

Question 1: How do you view the educational value of the VCS?	
Category	Criteria Guide
High	<i>Valuable / Useful / Helpful / Good / Interesting / Effective / Great / Nice ...</i>
Moderate	<i>Potentially / Fairly / Moderately / Somewhat / High theory but prototype limited / Good concept but need more work / Some value / Interesting concept ...</i>
Not sure	<i>Unsure / Don't know / Difficult to assess ...</i>
None	<i>None / Low / Minimal / Very limited ...</i>
N/A	<i>Missing or nonsensical</i>

favourably on the simulation system he developed (bias). The criteria guidelines developed for response categorisation may also be flawed, allowing for multiple interpretations. In order to minimise the chances of measurement errors, a number of raters were employed to individually judge responses against a common set of criteria. In total, five raters (the researcher, 2 PhD students and 2 academic staff in education) were employed to independently rate ~40% of the responses for each question using the criteria guidelines outlined in Tables 7.6 and 7.7 (the remaining 60% of data were rated by the researcher only).

While rater training is usually important for establishing inter-rater reliability (Creswell, 2008; Castorr et al., 1990), the nature of the data to be quantified in this study was not complex. Besides, the objective of using additional raters in this instance was to measure the reliability of the researcher's rating of responses. The researcher was therefore careful not to 'over-train' the raters, but rather use the category schemes as the main guideline for independent rating. The five raters' categorisation of responses was then analysed in order to identify any difficult items, i.e. responses where conflicting categorisation had taken place.

Chi (1997) suggests an inter-rater reliability of less than eighty percent 80% as having a great deal of discrepancy while other literature recommends 90-95% rater agreement (Poulson, 1987). Inspection of the rater dataset in this study showed that, in most cases, all 5 raters were in complete agreement. Table 7.8, which lists the rater agreement for each question, shows that the percentage of agreement

Table 7.7.: Categories and Schemes Established for Questions 2 and 3

Question 2: Do you believe the VCS is an effective means of training? How and Why?	
Question 3: Do you believe the VCS generates a valuable learning experience? How and Why?	
Category	Criteria Guide
Yes	<i>Yes / Yes at a basic level / Not for me but for others / Yes but can be improved further ...</i>
Potentially	<i>Not now, but with further work / If more features / More options / Can be in the future / Good concept ...</i>
Not sure	<i>Unsure / Don't know / Not enough time to make an opinion ...</i>
No	<i>No / Not so much / Not in this version ...</i>
N/A	<i>Missing or nonsensical</i>

between the 5 raters was ~97%.

Table 7.8.: Inter-Rater Agreement for Questions 1, 2 and 3

Question	Agreement
1	97%
2	97.6%
3	97.3%

60 responses rated for each question

Based on the inter-rater agreement findings, the complete set of quantitative data will be regarded as having 97% agreement in the analysis. The dataset from the quantification process, which was produced in SPSS TAS, was imported into the SPSS Statistics software and merged with data from the VCS HCI Survey for further quantitative analysis.

7.4.2. VCS Education Survey: Findings

The response frequencies in Tables 7.9, 7.10 and 7.11 are based on the quantitative datasets reduced from the qualitative responses to questions 1, 2 and 3, respectively.

7.4.2.1. Question 1

How do you view the educational value of the VCS?

In response to Question 1, a large portion of the population rated the educational value of the VCS as either ‘High’ (~62%) or ‘Moderate’ (~19%, Table 7.9).

Table 7.9.: Response Frequencies, Question 1

Category	Percentage
High	62.4%
Moderate	18.8%
Not sure	4.2%
None	9.7%
N/A	4.8%

N=159

One response that was rated into the ‘Moderate’ category reflects an important argument made by this research, namely that the VCS is not necessarily useful as a ‘stand-alone’ module. Rather, it should be used in the context of a guided exercise where training and follow-up are integrated and made relevant to the overall objectives of the teacher education course outline:

In conjunction with other tools and resources, it could be useful to look at factors in the classroom. (ST0007)

Several respondents pointed out that the VCS was useful in reminding them of the distinctiveness of each student:

Very useful for instilling an understanding of the fact that each student is unique in their dispositions, issues, abilities, motivation, agitation levels and this programme helps the new teacher to continually take this into consideration. (ST0016)

And the capacity of the VCS to challenge preservice teachers’ preconceptions was well addressed by this respondent:

It is quite interesting to see how the things you instinctively think will keep the attention of a class or a particular student, often don’t! Hence,

Chapter 7. VCS Qualitative Measures

this is a good tool to make you think about your own expectations and the reality of these in a real classroom setting. (ST0021)

The value-adding potential of simulation technology was also raised:

It offers a platform which other ways of education will not be able to offer for the educators to see how teachers, students, tasks and classroom environment interact with each other, and how this will influence the students' learning. (ST0045)

Other participants in the study were more sceptical, particularly in regard to the fidelity of the user interface, level of attribute detail or both:

Its great in theory, but not real enough. (ST0025)

...does not yet have enough attributes that make it a true enough simulation of a classroom environment. (ST0035)

In summary, eighty-one percent of the population felt the VCS was moderately to high in its role for education. Students acknowledged the value-added potential of the simulation in enabling learning.

7.4.2.2. Question 2

Do you believe the VCS is an effective means of training? How and Why?

In response to Question 2, nearly 70% found the VCS to be an effective means of training while another 8.5% felt that it had potential (Table 7.10).

Of those responses categorised as 'Potentially', some reiterated the need for the VCS to be integrated with other teaching and learning strategies:

VCS is one small part of (perhaps) a suite of effective means of training. It certainly has the potential to be morphed into different versions of this concept for different areas of training. (ST0016)

Table 7.10.: Response Frequencies, Question 2

Category	Percentage
Yes	69.7%
Potentially	8.5%
Not sure	5.4%
No	12.7%
N/A	3.6%

N=159

There were also those who felt that the VCS needs to be developed further for it to be more efficient:

At this stage I don't think it is, but it has a lot of potential once it has been further developed to become a very useful tool in pre-teacher training. The concept of it is really fantastic but it still needs a lot more fine tuning. (ST0035)

Many put extra emphasis on the VCS as an effective means of transferring theoretical knowledge and understanding, as indicated by this respondent:

I believe that it is incredibly effective from a theoretical perspective. Obviously, it is impossible to simulate the dynamics of a real, live classroom. However, it has already made me think about the different aspects of traditional STIC theory. (ST0057)

In summary, seventy percent of the population felt the VCS was an effective means of training. A further eight percent acknowledged its potential, provided further refinement of the software and/or the context in which it is used.

7.4.2.3. Question 3

Do you believe the VCS generates a valuable learning experience? How and Why?

Over 74% of the respondents rated the learning experience using the VCS (measured by Question 3) as valuable while a further 6.6% indicated that the VCS has the potential to do so (Table 7.11).

Table 7.11.: Response Frequencies, Question 3

Category	Percentage
Yes	74.5%
Potentially	6.6%
Not sure	6.7%
No	7.3%
N/A	3.6%

N=159

The VCS as a practical approach to learning theory received a number of favourable comments, e.g.:

Yes because it shows the theory in a practical way. (ST0101)

Yes [...]. We are learning in a more hands on (before the classroom/students) method and seeing physical results from our actions. (ST0133)

Other positive reactions to the learning experience involved a focus on the interaction that occurs between class entities (including the teacher) and their attributes:

Yes, as it teaches us to be aware of all students in the class. It teaches us to keep a balance in question answering and student management so that all students have a similar learning experience. (ST0093)

I think seeing the effect/ the correlation between questions, answers, learning, attention span and confidence was very helpful[...] (ST0091)

Yes. It effectively demonstrates the connection between knowledge state, self efficacy, metacognition, distractibility and the events affecting them. (ST0077)

Some respondents also highlighted the usefulness of the results (graphs) at the end of the lesson:

[...]And of course the other valuable end is seeing the output graphs and considering the individual requirements (such as Annelie answered a question incorrectly, we provided feedback and she learnt something, but her self-efficacy went down as a result of getting the question wrong). (ST0009)

In summary, seventy-five percent of the population felt that the learning experience of the VCS was valuable. A number of education students emphasised its ability to demonstrate student-teacher interaction and relevant theories.

7.4.3. Questions 4-18: Qualitative Analyses Through Double Extraction and Validation

The quantification activities employed for responses to questions 1, 2, and 3 in the VCS Education Survey were not replicated for questions 4 to 18. Rather, the remaining questions were analysed and interpreted using different analytic techniques for theme identification and response categorisation. While this too involved attributing a count/numerical value to question responses, individual responses were not rated to a fixed scale (a single response can belong to many categories). Moreover, the findings were not merged with the VCS HCI Survey data for statistical analysis, as was the case for questions 1, 2 and 3.

“Double extraction and validation” in the section title refers to the involvement of two software applications used for computerised text analysis (CTA) and reliability validation. As a general pre-processing procedure for all question responses, SPSS TAS was employed to automatically extract and create categories for significant terms. Subsequent use of SPSS TAS involved the manual task of creating, renaming and merging themes, as well as re-categorisation based on the researcher’s analyses of responses. In order to validate these interpretations, an objective approach similar to that used for the STIC Questionnaire (Section 7.3) was made possible with RapidMiner (see Section 7.2.2 for processes involved). RapidMiner was configured to report the most frequently used words used by the respondents. Where a single response made repetitive use of a word, that word was only given a frequency of 1. A stemming operator was used to reduce words to their root form (e.g. the tokens ‘distracted’, ‘distractibility’, ‘distracting’ etc. were

treated as the same word). Frequencies for words regarded as synonyms, or otherwise related in the given context, were merged. Validation with RapidMiner was only performed on responses to questions that lent themselves to word count, e.g. short and concise answers that readily identified specific aspects sought after by the question.

Creswell (2008) and Strauss and Corbin (1990) provide good overviews of the procedures involved in qualitative data analysis. A similar approach is adopted in this research, although responses to the VCS Education Survey were generally quite short (2-4 sentences). Thus, the extraction of categories and coding processes were arguably more straightforward than what is the case in Creswell's (2008) illustrated examples (which use interview transcripts and narrative passages).

7.4.3.1. Question 4

Question 4 of the VCS Education Survey reads:

*On which element/aspect of the simulation did you place most emphasis?
Please comment*

The visual entities (and their attributes) in the VCS graphical user interface featured prominently in the SPSS TAS analysis of responses to this item, where the main themes included 'student' (with related themes 'distractibility', 'self-efficacy' and 'knowledge') and 'question' (with related themes 'difficulty' and 'novelty'). A number of responses were categorised into several themes, e.g.:

*At first we placed most emphasis on the **students** we knew would answer the **question** correctly then chose to change the lower the level of **difficulty** and increase the level of **novelty** to see how students like Kennet, Sunniva and Maja would go, knowing they would be more likely to get the answer correct. We also wanted to see if it would increase their **confidence**. (ST0004)*

Similarly:

*Keeping the students **attention**. It felt like if you could keep all the **students** in the green, you were doing a good job. But when you look*

at the **results graph**, it shows some surprises. E.g. the student was paying attention, but not **learning** as much as another. It would be interesting to go back and do it again with a different **question** format to see if the students learnt better or worse. In real life there are no second chances, so I like this program! (ST0016)

Table 7.12 shows the most significant themes found in the responses to question 4. Some identified categories that did not receive a significant number of responses include mood, feedback, interest and metacognition. RapidMiner's count of the number of responses that included a particular word one or more times is provided in Table 7.13.

Table 7.12.: Response Themes and Categorisation, Question 4

Theme/Category	Number of Responses
Question(s) Task(s)	86
Student(s)	76
Student Distractibility Attention	61
Question Difficulty	34
VCS Graph Results	25
Question Novelty	15
Student Self-Efficacy	15
Student Knowledge	14
Answer	10

N=159

A comparison of Tables 7.12 and 7.13 suggests that there is little discrepancy between the two modes of analysis and mining. This is likely related to the fact that underlying themes in the responses explicitly match the words most frequently used.

Not surprisingly, student and question factors rate high in respondents' initial impression of the simulation. Two major themes that were not reported, though they are implied through these factors, are classroom interaction and management:

In order to effectively stimulate the students we found out that when the difficulty and the novelty of the question was at the medium level, students responded positively in relation to their self efficacy,

Table 7.13.: Word Frequency, Question 4

Word	Frequency
Question(s)	83
Student(s)	76
Distractibility Attention	69
Answer Response	27
Difficulty	25
Graph Results	29
Self-Efficacy	18
Knowledge	15
Novelty	15
Interest	13

N=159

distraction as well as understanding. When the difficulty was too high, the distraction in the class was also very high. (ST0036)

In summary, the education students gave much attention to the visual and interactive components in the VCS. A relatively small portion of the population made explicit reference to entity attributes and interaction-related variables. One can only speculate how these findings would compare to those from a group of expert teachers. Individuals with teaching experience may have been more inquisitive about the results charts, variables of interaction and how these compare to the real context. Many of the subsequent survey questions, including the next one, provided the education students an opportunity to reflect and respond to more specific details of the VCS.

RapidMiner was used to validate the analysis findings. The themes and their frequency ranking as per RapidMiner’s output correspond well with the analytic findings.

7.4.3.2. Question 5

Question 5 of the VCS Education Survey reads:

From your understanding of the VCS, what are the attributes of the students that the teacher need to take into consideration when posing a

task or question?

This question relates directly to student attributes in the student-teacher interaction context of the VCS. A number of responses, however, surpass student attributes that are actually present in the simulation. Responses like the one below suggest that the VCS is able to impart a perception of properties and interactions more advanced than what is specified in the simulation model.

students' prior knowledge, their understanding skills, their interests, their confidence, their enthusiasm, their patience, their physical condition, their engagement, their emotional state, their time management skills, their attitude towards learning, classroom environment, their cultural background, their home difficulties, students' level, their personalities, etc. (ST0045)

Table 7.14 shows the major categories that emerged from the analysis. Behaviour management, in terms of student distractibility, was ranked by many as an important factor.

Table 7.14.: Response Themes and Categorisation, Question 5

Theme/Category	Number of Responses
Student Distractibility Attention	81
Student Knowledge	48
Student Self-Efficacy Confidence	43
Student Ability/Skills Intelligence	38
Student Mood	24
Student Metacognition	7

N=159

This reflects a well highlighted feature/challenge when teaching in the VCS; students who are too distracted do not register questions asked to the class:

The short term issue seems to be distractibility [...] (ST0009)

Their distractibility is obviously of major concern; if the question is too hard some may not even take the challenge of thinking of an answer. (ST0055)

As shown in Table 7.15, RapidMiner yielded similar categories based on pure word count.

Table 7.15.: Word Frequency, Question 5

Word	Frequency
Distractibility Attention	91
Knowledge	48
Self-Efficacy Confidence	41
Ability/Skills Intelligence	32
Mood	24
Metacognition	7

N=159

In summary, all attributes included in the conceptual model of the VCS Student entity were acknowledged by the population (although some were not given as much emphasis). Major differences can be observed when comparing responses to the second question in the STIC Questionnaire (Section 7.3.2), which reads; “*What factors affect student learning in the classroom?*”. The VCS has been successful in highlighting a number of student properties that were not much considered, if at all, in the STIC Questionnaire. How behaviour affects student learning, for example, was only considered by 12 of 84 students (~14%) in the first questionnaire. In the current survey question, 91 of 154 students (59%) regarded student attention and distractibility as the most important factor to consider when posing a task.

Although the two questions compared are not identical and operate under different assumptions (one is general, the other VCS-specific), these findings nonetheless demonstrate an increased awareness of student factors and possible shift in thinking.

7.4.3.3. Question 6

Question 6 of the VCS Education Survey reads:

Can you list the attributes of a teacher, as presented either explicitly or otherwise in the VCS?

The VCS user takes on the role as the teacher. As such, no teacher attributes are explicitly modelled in the simulation system. This point was acknowledged by some respondents, e.g.:

VCS seems not to take into account the attributes of the teacher themselves, but rather the attributes of a small subset of the actions of the teacher, such as questioning attributes (novelty/difficulty), attention-getting, and idle-time. The actual attributes of the teacher (e.g. enthusiasm, content knowledge, etc.) seem to be much more difficult to model. (ST0005)

A number of responses to this question reflected back on the properties of the student and the many considerations the teacher should make with regard to these in the interaction context. This is reflected in the established themes (Table 7.16), where adaptive teaching, classroom management and personal qualities of the teacher are prominent.

Table 7.16.: Response Themes and Categorisation, Question 6

Theme/Category	Number of Responses
Adaptive Teaching (cater for all students)	95
Behaviour Management	60
Care and Compassion	28
Enthusiasm	24
Timing of Interaction and Wait-Time	17
Feedback	11

N=159

While knowledge of students' learning preferences and range of interaction and organisational skills were frequently mentioned, very few responses focused on the teacher's content knowledge. This could be related to the design of the Task entity in the VCS, which does not focus on specific subject content. The responses to this question did not lend themselves well to an objective word count in RapidMiner, as the most frequently used terms, e.g. 'student', 'question', 'knowledge', 'difficulty' and 'teacher', do not readily describe teacher attributes.

In summary, a large number of students felt that the teacher's capacity to cater for individual students was essential. Interesting observations can be made when comparing responses to those from Question 1 of the STIC Questionnaire (Section 7.3.1), which reads; "*Briefly identify at least three characteristics of a teacher that would be useful in the interaction in the classroom?*". Behaviour management, which

was ranked very highly in the VCS Education Survey, was not even reported for the STIC Questionnaire. Teacher knowledge and skills, which was ranked as most important in the first questionnaire, did not make the list in the current survey. Similar observations can be made when comparing other categories in the reports as well.

Again, it should be acknowledged that the two questions compared are not identical and operate under different assumptions (one is general, the other VCS-specific). The findings are nonetheless useful in providing some insight into how the population's perception of the teacher's role in the classroom may have been advanced and/or adjusted.

7.4.3.4. Question 7

Question 7 of the VCS Education Survey reads:

In posing a question or task to the class or a student, what does the teacher need to know to minimise distractibility and optimise learning?

Table 7.17 (prominent themes extracted from responses) and Table 7.18 (frequency of words that correspond to each category) make it clear that responses to this question re-enforce many aspects of what is modelled in the VCS.

Table 7.17.: Response Themes and Categorisation, Question 7

Theme/Category	Number of Responses
Question Difficulty	57
Student Knowledge and Understanding	56
Question Novelty	53
Student Interest and Motivation	29
Timing (and Wait-Time)	18
Student Self-Efficacy	17
Mood	9

N=159

A particular emphasis is placed on the interaction between student knowledge and question attributes as well as the teacher's knowledge of students' learning preference (interest and motivation).

Table 7.18.: Word Frequency, Question 7

Word	Frequency
Knowledge Understanding	47
Difficulty	43
Interest Motivation	27
Novelty	22
Efficacy Confidence	17
Time	15
Mood	7

N=159

In posing a question, the teacher needed to ask a question that every student understood, found interesting and could feel confident enough attempting. (ST0027)

It is interesting to note that, in answering this question, a number of respondents elaborate and expand on theories that are either explicitly or implicitly included in the VCS.

Ensure that the level of difficulty is not too far past their existing knowledge base, make sure that the student is comfortable enough and trusting of the teacher enough, to admit when they are unsure or don't know the answer to inspire dialogue on the topic, create understanding and therefore optimise learning and simultaneously minimise distractibility due to the high level of engagement during the dialogue. (ST0004)

In summary, students agreed that learners' knowledge and question difficulty (and the interaction between these) are the most important factors to consider when posing a task. A comparison with question 3 in the STIC Questionnaire (Section 7.3.3) shows very similar ranking of these properties.

7.4.3.5. Question 8

Question 8 of the VCS Education Survey reads:

What are the strengths of the VCS? Please give specific examples.

Chapter 7. VCS Qualitative Measures

This question provided users with an outlet to describe what they felt were the overall strengths of the VCS. A large number of responses were very descriptive and spanned across a number of strengths.

It enables you to simulate a classroom for a short or long amount of time and then reflect on it, analyse the results, try out new techniques and see how students respond, compare analyses, and keep on trying to explore what skills work and don't in this environment. (ST0081)

Comments like these allowed for the establishment of more elaborate themes than what was the case for previous questions in the survey. Table 7.19 lists the four themes that were commonly encountered in the data.

Table 7.19.: Response Themes and Categorisation, Question 8

Theme/Category	Description	Number of Responses
Theory	Focus on the VCS' educational theories involving student-teacher interaction, entities and their attributes.	60
Practice Experiment Observe Evaluate Reflect	Focus on how the VCS provides (repeated) hands-on practice and instant feedback in a "safe environment" (ST0012) that allows users to explore, "experiment" and "manipulate" (ST0158), "fine-tune delivery techniques" (ST0020), observe, "compare analyses" (ST0021), discuss and reflect.	59
Results Feedback	Focus on the results (charts) provided in the VCS.	43
Classroom, Lesson and Behaviour Management	Focus on time management, student behaviour management (distractibility) and organisational skills.	34

N=159

A number of responses were categorised into more than just one theme. The comment below, for example, was fitted into the 'Practice, Experiment...' and 'Results/Feedback' categories:

Provides a somewhat 'scientific' means to understanding students in a classroom. This is achieved through the results section. Teachers are able to reflect on the whole lesson and observe accurate results of each student. Another strength is that it is a virtual classroom, so trial and error while training does not impact real students just virtual ones! (ST0110)

The graphical user interface did also receive some positive remarks (e.g. “It’s a pretty interface - 10/10 for that”, ST0100) and the strengths of the results charts were frequently implied. The following respondent, however, was more explicit than most:

The graph plotting each student’s knowledge, confidence, etc. was brilliant - i like how it plots each individual students progress, even if i do not ask them questions/get their attention. (ST0102)

An important aspect of the VCS design is the mapping to wall-clock (real) time. Judging by a number of responses to this as well as other questions, this added strength to the observability and real-life mapping of states and events.

I like the time management aspect (that its in real time). It is also good that you can see the student faces and the changing expressions to help with your judgement. (ST0072)

To encourage reflection is a key aim of the VCS. One respondent reminded the researcher that the survey itself was also useful for reflective practice.

The main strength of the program is that it has made me think about the balance of a classroom that i may not of thought about before. The addition of this survey definitely made it all a lot more clear to me because i had to actually look back on my results and make connections from them. (ST0034)

In summary, a number of education students highlighted that the simulation allows for an experiment—observe—reflect cycle in a safe environment. By collapsing the two top ranked themes, the VCS’ main strength appears to be praxis; e.g. its capacity to unite theory, reflection and practice.

7.4.3.6. Question 9

Question 9 of the VCS Education Survey reads:

From your knowledge and experience in classrooms, does the simulation meaningfully resemble the real environment? Please comment.

Structurally, this question seeks a ‘yes’/‘no’ response with further elaborating statements. When considering the implications of the question, however, responses were not expected to be as clear-cut as that. This was confirmed by processing the dataset with RapidMiner, in which the stopword operator was disabled to allow the words ‘yes’ and ‘no’ be included in the output. 36 and 30 occurrences of the words ‘no’ and ‘yes’ respectively were counted. The established themes and response categorisation in the analysis of this question can be observed in Table 7.20. A quick scan through the collected data suggested that respondents were divided on this subject, with many finding it difficult to give a straight answer. One student simply answered “yes and no!!” (ST0021), another wrote “70%” (ST0023). Both of these responses were fitted into the ‘Yes | To Some Extent’ category.

Table 7.20.: Response Themes and Categorisation, Question 9

Theme/Category	Number of Responses
Yes To Some Extent	84
Unsure	16
No	46
N/A	13

N=159

A number of respondents felt that they could not comment because they did not have any teaching experience in the real classroom (‘Unsure’ category):

I haven’t had any teaching experience yet, so can’t really say. From my memories of a student, it’s reasonably accurate though obviously simplified. (ST0024)

Of those who answered ‘yes’, many commented on the VCS’ resemblance to certain aspects/scenarios of the real environment.

In some cases yes. In a class such as English, maths, science etc. where questions tend to dominate the class time it is quite useful. For more interactive classes such as music, PE, performing arts it doesn't allow for the students to interact with each other. (ST0083)

Over 40 people said that the VCS does not meaningfully resemble the real environment. This does not, however, necessarily mean that the experience was without value, e.g.:

Well no, because maintaining student attention isn't as easy as a click of a mouse, but it is based on the general principles of the classroom which is a good tool to get you to think about them. (ST0069)

No because there aren't enough students - but as an introduction tool to cognition, etc. it is effective. (ST0087)

In summary, considering that this question was posed to novice teachers with little or no teaching experience, responses cannot be used to estimate how well the VCS actually resemble the real environment. The reduced form of the qualitative data does nonetheless provide an insight into the education students' perception of the VCS as compared to the real world. The response categories do not gauge the usefulness of the simulation, e.g. a 'no' does not mean that the VCS is not a valuable tool for learning.

7.4.3.7. Question 10

Question 10 of the VCS Education Survey reads:

What improvements need to be incorporated into the VCS as a teaching and/or training tool?

This question yielded a number of suggestions for improving the VCS. As shown in Table 7.21, a common theme involved a limited range of student-teacher interactions available in the simulation.

Some suggestions, however, are arguably beyond the scope and core aims of the VCS, e.g.:

Table 7.21.: Response Themes and Categorisation, Question 10

Theme/Category	Number of Responses
More Options for Student-Teacher Interaction (not just Q&A, other modes of learning, group discussion, interact with one student only)	34
User Interface (sound animation)	27
More Students in Classroom	18
Real Questions	17
More Student Attributes (e.g. personality, learning preferences, students with disabilities, ethnicity)	16
Add Teacher Attributes in Software (e.g. personality, tone of voice, positioning in classroom)	7
Enhance Results (e.g. side-by-side comparison of charts, feedback on teacher's performance, textual summary)	5
Student-Student Interaction	5
Better Guidelines for Using the VCS (e.g. objectives, purpose, outcomes)	5
NA (e.g. blank, "can't think of any", "good as-is")	39

N=159

I think it would be good if education students were able to choose from a number of other activities, not just asking questions. For example, if you could go away, develop a mock lesson plan, come back to the VCS and then put the students through a number of activities such as 10 mins on a worksheet, 10 mins on a video, 10 mins on primary source analysis and so on, to more accurately reflect the diversity of activities that teachers do in a classroom. (ST0038)

The following person provided a number of useful suggestions:

Examples of terms used in the class room could be an added bonus. All teachers like to be introduced to new ways of regaining attention. I think that the focus on distractibility of students should be broadened to encompass the other areas of knowledge, metacognition, self efficacy etc. etc. Perhaps the program could be developed into a broader game-style activity, where over a number of lessons you can improve students levels

of understanding? It'd also be good to understand more about why the distractibility goes up or down at times when the student is not being 'interacted with'. The distractibility meter is handy for the activity but meters for the other measured areas could also be included. You could change the focus between distractibility and knowledge to further highlight the importance of behaviour management and its impact on learning. (ST0047)

In particular, the above notion of adding expert exemplars of teacher dialogue with the class (e.g. for discipline and reward) could provide a useful resource for preservice teachers.

References to the user interface were mostly related to the visual and audible representation of students. A number of users would like to see more detail in terms of student animation (body language and facial expressions) and a classroom view more akin to real life.

I am not suggesting that this need be like a PS3 game. However, there is quite a level of detail in such games that allow for 1) more interest, 2) more choice on behalf of the user, which have different results. 3) audio and greater visuals give a greater impression of reality and would simulate the learning environment better. Get some more funding from the government and find a graphic designer? This has potential to be widely used globally. (ST0072)

Making the question entity generic (e.g. no textual representation, nor any relation to a subject area) was a significant design decision in this study. 17 survey respondents highlighted either directly or indirectly that questions should be real and tied to a subject area, e.g.:

More specific questions relating to different subjects, and different reactions from the students. (ST0067)

The option of choosing a topic (e.g. climate) and asking questions relevant to that topic. The options can be limited but will make the activity more meaningful to us as teachers. (ST0097)

In summary, the responses provided useful feedback for the future direction of the VCS. The research had already identified a number of the areas for improvement that surfaced in the survey. Nonetheless, the users' input has added weight to what should be allocated the highest priority. More than 10% of the respondents asked for more students in the classroom and that the questions were tied up to specific subject areas. Although these requests disagree with a design decision that was carefully considered for this research, they are relevant and will be considered in future development.

7.4.3.8. Question 11

Question 11 of the VCS Education Survey reads:

Explain how knowing about the student's prior knowledge about the topic is important for the teacher?

Although this question was not explicitly related to the VCS, a number of responses were influenced by the simulation experience.

Entering the simulation, I wasn't aware of the students' prior knowledge. If I had, it would have greatly affected the difficulty and the novelty of questions, as well as to whom some questions would be directed. (ST0104)

As explained monitoring the VCS student response and getting the right balance of question is key, therefore to set the right level of question in terms of difficulty and novelty is extremely important. Prior knowledge affects how difficult the question will be for the student. (ST0158)

Nearly all respondents provided reasons that were on target, although the level of depth in their reasoning varied. Table 7.22 provides an overview of how relevant the students' responses were to the question.

Answers suggested that prior knowledge is important for knowing at what difficulty level questions should be pitched, building new knowledge, keeping student engaged, maintaining confidence, reducing distractibility and making learning more meaningful.

Table 7.22.: Response Themes and Categorisation, Question 11

Theme/Category	Number of Responses
Good Explanation (e.g. one aspect explained well or listed 2 or more reasons)	104
Short Explanation (e.g. brief mention of 1 valid reason without explaining)	42
Nonsensical (did not address the question)	3
NA (blank or otherwise, e.g. “ran out of time”)	10

N=159

7.4.3.9. Question 12

Question 12 of the VCS Education Survey reads:

Is self-efficacy affected by question answering?

This appeared to be a straightforward question upon which nearly all respondents agreed (see Table 7.23).

Table 7.23.: Response Themes and Categorisation, Question 12

Theme/Category	Number of Responses
Yes	145
Unsure (“maybe”, “not sure”, not familiar with “self-efficacy”)	4
No	1
NA (blank or otherwise, e.g. “Ran out of time”)	9

N=159

A number of people provided further elaboration and insight beyond ‘yes’/‘no’, e.g.:

Yes. Self-efficacy should be affected by whether or not a student is picked for answering the question, whether the answer they give is right or wrong, and whether the question asked was too easy, too hard, or at a level on par with the students current knowledge. If they have the

Chapter 7. VCS Qualitative Measures

question wrong, self-efficacy could also be affected by how the teacher answers the question, and their attitude towards the student and their error. (ST0005)

The one student who answered “no” to this question was not necessarily ignorant to the fact that self-efficacy *could* be affected by question answering:

No it isn't because that will not change from a 5 minute span. (ST0087)

In summary, the findings from this question are very positive. In the VCS trials, which were all administered by the researcher, a number of participants would ask the researcher what the word “self-efficacy” meant. Only a single response to the STIC Questionnaire used the term. Responses to this survey question indicate that a number of education students may not only have learnt a common term in education theory, but also how it operates in the classroom interaction context.

7.4.3.10. Question 13

Question 13 of the VCS Education Survey reads:

How could students' level of prior knowledge about a lesson topic be related to their level of distractibility?

A huge majority explained this question in terms of the interaction between prior knowledge, subject difficulty and distractibility (see Table 7.24).

Table 7.24.: Response Themes and Categorisation, Question 13

Theme/Category	Number of Responses
Interaction between prior knowledge and subject difficulty	136
Unrelated Nonsensical	12
NA (blank or otherwise, e.g. “Ran out of time”)	11

N=159

Most responses can be summed up as follows; if a question does not challenge a student’s prior knowledge, or if the challenge is too great, the student will lose

interest and become distracted. The following response is representative of a large portion of the answers given:

If they feel overwhelmed by new information they may give up and switch off. If it is too easy and they already know it they may become bored. (ST0042)

7.4.3.11. Question 14

Question 14 of the VCS Education Survey reads:

Explain how consideration of the level of difficulty of a question is important for a teacher.

Respondents provided a variety of reasons as to why the question difficulty demands careful attention from the teacher. Answers with similar arguments were merged into the broader themes as shown in Table 7.25.

Table 7.25.: Response Themes and Categorisation, Question 14

Theme/Category	Number of Responses
Previous Knowledge Learning Build and Gauge Knowledge	77
Distractibility Bored Engaged Interest	72
Self-Efficacy Mood	38
Unsure Nonsensical	8
NA	10

N=159

Obviously we need to offer questions that students can actually answer, at least some of the time, this builds morale and engagement may follow. On the other hand very difficult questions still require answers and some students may get satisfaction from grappling with them even if they are unable to reach an answer. How one balances this mix of requirements is part of a teacher's style and may be significantly linked to other less tangible factors such as their popularity as individuals, the nature of the

Chapter 7. VCS Qualitative Measures

learning environment, the class' overall interest in the subject, the mood of the class as an aggregate, or even the mood of dominant individuals, at the particular time of delivery. Students thrive on challenge, but have little resilience in disappointment, managing all of these factors helps the teacher to negotiate this difficult aspect of the questioning process. (ST0020)

The 'Previous Knowledge' category includes responses that involve question difficulty as an instrument to measure and/or improve student learning and knowledge, e.g.:

The level of difficulty is linked to developing higher order thinking. It is important to ask these questions, but they must be scaffolded along the way before reaching them. (ST0107)

Responses fitted into the 'Distractibility' category make references similar to those provided to the previous question; if the challenge does not correspond to the students' level of knowledge, they will not be as engaged, get bored and/or lose interest. How a question's difficulty may impact on a student's mood and/or self-efficacy was also brought up:

Question difficulty has the potential to affect the mood, distractibility, self-efficacy and ability to add to the knowledge of the student. Again, the art of the teacher is to present questions at a difficulty appropriate to the knowledge of the entire class, and to attempt to include the whole class despite differences between the knowledge of the individual students. (ST0005)

In summary, responses to this as well as other questions demonstrate that the education students were able to appropriately connect their responses to (and sometimes extend) the theories and practices promoted by the VCS. The notion of instructional scaffolding (e.g. Palincsar, 1986) to optimise cognitive performance was frequently implied by the respondents.

7.4.3.12. Question 15

Question 15 of the VCS Education Survey reads:

Annelie had a high level of distractibility after about 1 min into the lesson. Why do you think her level of distractibility increased and stayed quite high for the rest of the lesson?

Table 7.26 lists the themes extracted from this question as well as the number of responses fitted into each of these.

Table 7.26.: Response Themes and Categorisation, Question 15

Theme/Category	Number of Responses
Questions Too Easy Not Challenging Enough to Engage Her	88
Learning Preference Personality Sunniva Classroom Climate	31
Teacher Failed to Intervene Cater for Her Needs	27
Questions Too Difficult	12
Annelie Was Not Interested in the Subject	10
Unsure Not Related to Question	14
NA	16

N=159

What was expected from the question was for the respondents to make a connection between details in Annelie's student report card as well as her performance/behaviour in the simulation. A number of answers did reflect this well by correctly indicating that Annelie's prior knowledge was high (thus questions may not have sufficiently engaged/challenged her) and that her best friend Sunniva could be a bad influence on her.

Annelie has quite a high cognitive level and therefore may find the questions/tasks too easy thus leading to distraction. She may need to be given some more challenging material. Also, she needs to be moved away from her friend (Sunniva) who she gets distracted by. (ST0124)

The teacher's failure to intervene (e.g. discipline) and/or cater for her learning preference was also suggested as possible reasons.

Possibly because the teacher did not get her attention and maintain it through asking her questions and involving her in the lesson. (ST0051)

Chapter 7. VCS Qualitative Measures

Although Annelie is probably the smartest kid in class, to propose that the questions were too difficult for her is not necessarily incorrect (her academic capabilities do have a limit).

She might not have been called to attention and the questions might have been too hard for her to follow since she was already only half paying attention. (ST0071)

The VCS does not take into consideration the subject matter taught. Nor does it include a student attribute to indicate that some lessons are more interesting than others. Hence, the theme ‘Annelie Was Not Interested in the Subject’ does not relate directly to what is actually present in the simulation model. It is nonetheless positive to observe that the VCS is able to inspire a level of thinking and discussion that supersedes the level of complexity in in the model.

Annelie has difficulty engaging in lessons overall, she tends to become distracted easily as she lacks a personal commitment/connection to school overall and to the material being discussed. She is the classic example of the student who slips through the cracks of our education system as strategies to maximise her involvement may be at the expense of other students or the class as a whole (purely in terms of time). (ST0020)

In summary, it is evident that the vast majority of the education students were able to connect what was happening in the VCS classroom with relevant theory and practice. This question was related to a specific scenario in the VCS. A number of responses described causal factors that are valid, yet not explicitly included in the system. This suggests that the system is able to demonstrate behaviours beyond what is manifested in the simulation model specification.

7.4.3.13. Question 16

Question 16 of the VCS Education Survey reads:

What is Kennet like as a student?

The VCS provides a report card for each student (all of which can be found in Appendix K). Kennet’s description reads:

Kennet is the class clown. Always happy and cheerful, his class presence can be both inspiring and annoying. Kennet has dyslexia, a rather low cognitive level and has a tendency to avoid any type of learning. Nonetheless, as long as he is not distracted by other things, he is always active in class and is happy to interact and take a punt on questions at any level of difficulty. There is reason to believe that his confidence is only skin-deep...

Table 7.27 indicates that the vast majority of the respondents picked up on Kennet's traits, as informed by the report card, lesson observations or both.

Table 7.27.: Response Themes and Categorisation, Question 16

Theme/Category	Number of Responses
Easily Distracted	93
Limited Knowledge Low to Medium Achiever	36
Happy Cheerful Friendly	28
Good Student Engaged	17
Confident	8
Diffident	6
Unsure Nonsensical	10
NA	8

N=159

A joy and a frustration. His happiness sparks the moral of the class but his tendency to avoid any kind of learning and his lower cognitive level make him a student who requires more scaffolding and teaching. (ST0017)

Kennet is a happy student, and willing to learn if the subject matter relates to him and he finds it interesting. but as his self confidence is low, he will be the class clown to hide behind that, he may also be hiding the fact that he is unsure about a certain topic. The teacher will need to pay attention to Kennet, and make sure he is understanding the content. (ST0112)

Again, the survey data demonstrated that a number of education students displayed an advanced level of thinking and a tendency to attribute personality traits further to what is modelled in the VCS.

The description of Kennet was that he was a class clown and sometimes distracted others. I think he is pretty smart though, loves attention. (ST0153)

Kennet can be a positive influence in the class because of his light hearted nature. However it is important to keep his confidence up so that his behaviour is not just a reflection of low self-esteem. He is diligent when engaged and his engagement would be a reflection of the difficulty and novelty of what is being discussed. (ST0103)

7.4.3.14. Question 17

Question 17 of the VCS Education Survey reads:

If Maja's face showed less sadness during the lesson, why would this be?

A majority of the education students cited that Maja's good performance, either by directly answering a question correctly or knowing the answer, had an impact on her mood (see Table 7.28). Many of those who did not make this direct connection suggested that a boost in self-efficacy, knowledge or engagement in class (often a consequence of question performance) had a positive effect on her mood.

A relatively high number of responses in the 'Nonsensical' category suggests that this question might have been open to misinterpretation or difficult to understand (answers like "*Because she's happier*" (ST0043), "*No, but she is quite emotional*" (ST0096) and "*She feels tired*" (ST0119) were all observed in this category). Some responses in this category may also be related to the education students being cautious in inferring from moods and facial expressions. A high number of answers fitted into the 'NA' category (e.g. blank, "pass" or "ran out of time") indicates that many respondents were pressed for time and unable to complete the survey by this stage. This could have impacted on the quality of responses as well as careful reading of the question text.

Table 7.28.: Response Themes and Categorisation, Question 17

Theme/Category	Number of Responses
Answered Question Correctly Knew the Answer	72
Increased Confidence Knowledge Learning	66
More Engaged	33
Interested in the Subject	12
Other factors	8
Nonsensical	14
NA	16

N=159

7.4.3.15. Question 18

Question 18 of the VCS Education Survey reads:

From your understanding of the VCS, in what scenario(s) is a student's Knowledge State affected?

This was the final question in the VCS Education Survey. Judging by the increase of responses fitted into the 'NA' category, many were unable to complete due to time restraints. Looking at the findings in Table 7.29, a surprisingly large number failed to demonstrate a reasonable explanation of how the knowledge state is affected in the VCS.

Table 7.29.: Response Themes and Categorisation, Question 18

Theme/Category	Number of Responses
Incorrect Nonsensical	52
Teacher Feedback to Incorrect Response	50
Learning From Another Student's Correct Answer	46
Unsure	7
NA	18

N=159

This is not to say that all 'Incorrect' responses fitted in the category did not demonstrate an understanding of some sorts, on the contrary:

Chapter 7. VCS Qualitative Measures

The level of the questions that are given to them, and whether or not there is a gradual but constant increase in the difficulty and novelty of the questions. (ST0050)

When the right balance between metacognition, cognitive ability and self efficiency is found for the student. (ST0034)

I thought that as the students answered questions correctly, then their knowledge state increased, but often this was not the case. (ST0027)

Some of the ‘Nonsensical’ responses may be attributed to respondent fatigue and/or time factors.

Despite these anomalies, the overall majority were still fitted into the two categories representing correct responses.

When students get questions wrong, intrigued by a wrong answer, they wonder what the right answer is. Thus the teacher feedback on a wrong answer affects the knowledge state in the positive. (ST0091)

When a question is answered correctly by someone else, most effectively when teacher feedback is given. Obviously if they answer the question themselves their knowledge state is unaffected. (ST0020)

7.5. Limitations

Only 50 minutes were available per tutorial session to allow for a user trial and collection of data. A number of students indicated that spending more time with the simulation would be of value. The limited time available for the VCS trials (a five minute lesson) may thus have affected the participant’s impression of the VCS.

[...]This program would need to be used over a extended period of time that would provide individual with more of an opportunity to feel the Virtual Classroom. (ST0085)

I think using it over a number of sessions would generate a valuable learning experience. (ST0010)

The VCS Education Survey had the highest number of participants, which may be attributed to the fact that it was run immediately after the trial had ended. The fact that the survey was administered online meant that students could complete the survey at a later time (e.g. from home), which some did. The online nature of the survey may also have improved the completion rate in the session, although there is no evidence to support this. Despite these positives, the limited time available disadvantaged the quality of responses, particularly for the later questions in the survey. This is also evident in the HCI Survey, which was administered immediately after the VCS Education Survey. The completion rate dropped from 159 for the VCS Education Survey to 86 for the HCI Survey.

7.6. Summary

Observations from the trials and responses to the VCS Education Survey suggest that the Virtual Classroom Simulation was an engaging experience that left many to ponder on the theories and practices of student-teacher interaction in the classroom. It also fostered a curiosity about the functional parameters of the VCS.

I would like to understand the model its based on a bit more to understand the value, i.e. the behaviours and 'feelings' of the students - how complex is this? (ST0013)

I would like to understand better how the system arrives at the behavioural simulations. (ST0114)

This chapter discussed the analyses and validation techniques implemented for the STIC Questionnaire and the VCS Education Survey. The STIC Questionnaire was designed to elicit concise terms. It did not provide the respondents with the opportunity to demonstrate their understanding through further elaboration. However, in comparing the categories established from the STIC Questionnaire

with those from the VCS Education Survey, there may be evidence to indicate that there has been an adjustment in their understanding of STIC and PCK.

Responses to three questions from the VCS Education Survey were quantified. Rater bias was not evident in the analysis of the inter-rater reliability, which involved five raters. Analysis of other qualitative data was assisted by SPSS TAS. Where applicable, RapidMiner was employed as an ‘objective rater’ to validate the researcher’s analytic observations. The researcher’s identification of themes and categorisation of responses yielded similar results to what was achieved using word frequency count in RapidMiner. No evidence of rater bias was found. This study has demonstrated that, for short concise responses, text mining software such as RapidMiner may be useful in monitoring rater reliability.

The STIC Questionnaire and the VCS Education Survey make part of the triangulation mixed methods design (Creswell, 2008, p. 557) of this study. Thus, some of the data and results reported in this chapter will be interpreted in conjunction with the data and results from the quantitative HCI Survey. All the data, qualitative and quantitative, provide a rich set of information that will be used to address the third research problem of this study (see Chapter 10). In the next two chapters, instruments involved in the collection of quantitative data will be analysed and reported.

Chapter 8.

Computer Familiarity Scale

Establishing acceptable levels of construct validity is critical, particularly when the measured constructs are to be further used for structural equation modeling.
(Segars and Grover, 1993, p. 518)

It would be difficult to assess the usefulness of the VCS without sufficient knowledge and understanding about participants' experience and attitudes towards computer technologies. Since there are several factors that could impact on a user's attitude towards the educational usefulness and/or technical aspects of the VCS software, this knowledge becomes important when analysing response data. The VCS HCI (Human-Computer Interaction) Survey component of this research study is therefore composed of two different instruments; one that measures users' computer familiarity and one that focuses on the user interaction satisfaction. This chapter is concerned with the former of the two instruments while Chapter 9 provides details on the latter.

In order to learn more about the users' familiarity with computers the study employed an already established instrument. Supervisors in education suggested an adaptation of a particular instrument used by the Organisation for Economic Co-operation and Development (OECD) in the PISA (Programme for International Student Assessment) 2000 Study (Adams and Wu, 2002) as highly suitable. The magnitude of the PISA 2000 study implied that the instrument would be well established, as well as valid and reliable. In fact, parts of the PISA 2000 Computer Familiarity instrument were adapted from an instrument

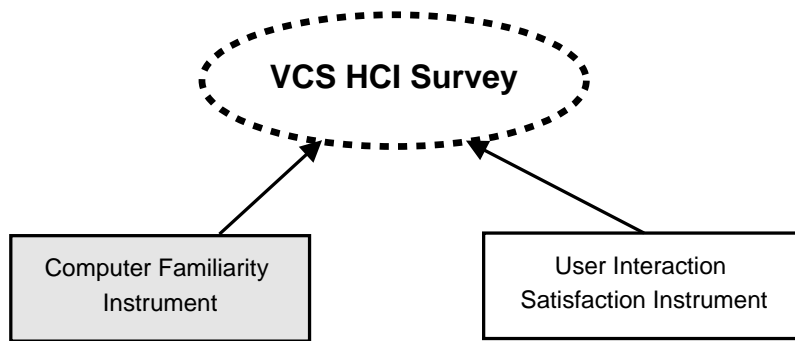


Figure 8.1.: VCS HCI Survey Composition

developed by the Educational Testing Service as it was “proved to be short and efficient” (Adams and Wu, 2002, p. 38). However, the PISA 2000 Study was designed for 15-year-olds in an international context. Some items in the questionnaire sub-scales were therefore modified from their original form to better suit this study and its participants. Thus, in order to assess the reliability and validity of the instrument in the context of this research, some parts of the instrument were put through further analysis and validation. This chapter provides more details about the scale, adaptations made, the validation process and a summary of findings.

8.1. Data

The two instruments, Computer Familiarity and User Interaction Satisfaction, were administered as the ‘VCS HCI Survey’. All questions in this survey were mandatory in that every item on a page had to be answered before the respondent could advance to the next screen. Inspection of the data, however, uncovered a number of problematic submissions. For example; some responses were incomplete (e.g. a respondent would exit the survey before completing all screens) while others had rated all items at one extreme end only. Therefore, 86 of the 110 collected survey submissions remained after data clean-up. The VCS HCI Survey was the last activity in the tutorial session. The relatively low number of submissions when compared to the preceding VCS Education Survey (see Chapter 7) may be attributed to respondents running out of time. This would also explain

why some surveys were only partly completed.

The survey was administered online, thus all information was captured directly into an electronic format. Collected data were exported from the survey software into the Microsoft Excel format and subsequently into a number of formats suitable for further analysis.

8.2. The Instrument

This study engaged a number of items adapted from the Computer Familiarity Instrument of the PISA 2000 Study. The complete instrument, which was available for download on the OECD website (OECD, 2000), was adapted for use in this study. The PISA 2000 Technical Report (Adams and Wu, 2002, pp. 242-243) accounts for most, but not all of the instrument's sub-scales. The four main sub-scales of the instrument are shown in Figure 8.2 and will be discussed in detail in the subsections below.

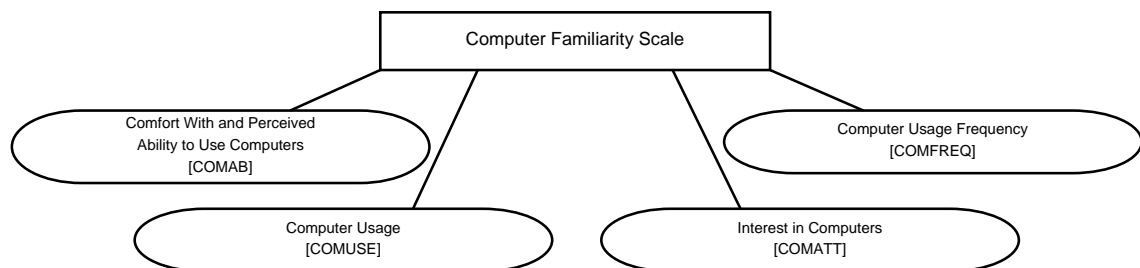


Figure 8.2.: Computer Familiarity Scale

Screen-shots of the final representation and layout of the questionnaire used in this study are available in Appendix S, with the scales for this particular instrument displayed in Figure S.3 on page 376. A capture of the original version of this instrument was included in Figure 6.9 on page 151. It may also be useful to refer to the code book for this instrument, which can be found in Appendix T.

8.2.1. Comfort With and Perceived Ability to Use Computers Scale (COMAB)

Table 8.1 accounts for the original sub-scale as implemented by the PISA study, while Table 8.2 provides the modified scale used for this research.

Table 8.1.: PISA Sub-Scale of Comfort With and Perceived Ability to Use Computers

<i>How comfortable:</i>	Item Code
are you with using a computer?	IT02Q01
are you with using a computer to write a paper?	IT02Q02
would you be taking a test on a computer?	IT02Q03
If you compare yourself with other 15-year-olds, how would you rate your ability to use a computer?	IT03Q01

Table 8.2.: VCS Sub-Scale of Comfort With and Perceived Ability to Use Computers

<i>How comfortable are you with using a computer for the following tasks?</i>	Item Code
Using a computer in general?	CMF_GEN
Using a computer to write a paper?	CMF_PPR
Taking a test on a computer?	CMF_TST
If you compare yourself with others in your course, how would you rate your ability to use a computer?	ABILITY

For the first three questions of the COMAB scale, a four-point scale was used with the response categories “very comfortable”, “comfortable”, “somewhat comfortable” and “not at all comfortable”. For the last item (IT03/ABILITY), a four-point scale was used with the response categories “excellent”, “good”, “fair” and “poor”. The adapted version of the PISA scale used the same scales as the original.

8.2.2. Computer Usage Scale (COMUSE)

The PISA sub-scale of computer usage, as shown in Table 8.3, consists of six questions. A five-point scale was used with the response categories “almost every day”, “a few times each week”, “between once a week and once a month”, “less

than once a month” and “never”. The PISA-derived VCS scale (e.g. adapted version, as shown in Table 8.4) consists of five questions and uses the same response categories as the original, with the exception of “almost every day” which was changed to “daily”.

Table 8.3.: PISA Sub-Scale of Computer Usage

<i>How often do you use:</i>	Item Code
the computer to help you learn school material?	IT05Q03
the computer for programming?	IT05Q04
<i>How often do you use each of the following kinds of computer software?</i>	
Word processing (e.g., Word® or Word Perfect®)	T06Q02
Spreadsheets (e.g., Lotus® or Excel®)	T06Q03
Drawing, painting or graphics	T06Q04
Educational software	T06Q05

Table 8.4.: VCS Sub-Scale of Computer Usage

<i>How often do you use each of the following kinds of computer software?</i>	Item Code
Games	SFRQ_GMS
Office related (e.g. Word Processing, Spreadsheets)	SFRQ_OFF
Drawing, painting, video or graphics	SFRQ_MED
Email client	SFRQ_EML
Internet web browser	SFRQ_WEB

A comparison of the two sub-scales shows that the VCS scale excluded the first two questions (IT05Q03 and IT05Q04) of the PISA scale. Further, with regard to computer software use, a few changes were made in the VCS scale. Specifically, the alterations included the introduction of three new items; “games”, “email client” and “Internet web browser”. The “educational software” item was removed while “word processing” and “spreadsheets” were merged into a single item “office related”. Finally, the item “drawing, painting or graphics” was altered to include “video”.

The modifications to this scale were made in order to better target the population of this study and the research objectives. In particular, the “games” item was deemed an important one for inclusion because the VCS lends itself to comparison

with various simulation-type games. The last four items aim to provide an indication of participant usage of software for work/study, creative and online purposes.

8.2.3. Interest in Computers Scale (COMATT)

As with the abovementioned sub-scales, participants' interest in computers could also affect their views on the VCS. The original PISA sub-scale is shown in Table 8.5.

Table 8.5.: PISA Sub-Scale of Interest in Computers

	Item Code
It is very important to me to work with a computer.	IT07Q01
To play or work with a computer is really fun.	IT08Q01
I use a computer because I am very interested in this.	IT09Q01
I forget the time, when I am working with the computer.	IT10Q01

Table 8.6.: VCS Sub-Scale of Interest in Computers

	Item Code
The computer is a very important tool to me.	CMP_IMP
To play or work with a computer is really fun.	CMP_FUN
I am generally very interested in computer technology.	CMP_INTR
I forget the time when I am working with the computer.	CMP_TIME

A two-point scale was used for all four items in this scale, with response categories “yes” and “no”. The VCS scale for interest in computers, as shown in Table 8.6, has kept all four items although some of these have been reworded.

8.2.4. Computer Usage Frequency Scale (COMFREQ)

The fourth sub-scale used by this study was not covered in the PISA 2000 Technical Report (Adams and Wu, 2002, pp. 242), but can be found in the “Computer Familiarity Questionnaire” available on the OECD website (OECD, 2000). The original scale (Table 8.7) concerns the frequency of computer usage “at home”, “at school”, “in the library that you use” and “at another place”.

Table 8.7.: PISA Sub-Scale of Computer Usage Frequency

How often do you use a computer:	Item Code
at home?	IT04Q01
at school?	IT04Q02
in the library that you use?	IT04Q03
at another place?	IT04Q04

Table 8.8.: VCS Sub-Scale of Computer Usage Frequency

How often do you use a computer at the following places?	Item Code
At home?	CFRQ_HOM
At school?	CFRQ_SCH
In the library that you use?	CFRQ_LIB
At another place (e.g. laptop, Internet cafe, etc.)?	CFRQ_OTH

The same question categories were reproduced in this study, with the exception of the last item which was altered to read “at another place (e.g. laptop, Internet cafe, etc.)”. Similar to the PISA Computer Usage sub-scale (COMUSE, Section 8.2.2), this scale used a five-point scale with the response categories “almost every day”, “a few times each week”, “between once a week and once a month”, “less than once a month” and “never”. The adapted scale (Table 8.8) uses the same scale as the original with the exception of the “almost every day” response category, which has been changed to “daily”.

8.3. Scale Structure Analysis

Section 6.7 in Chapter 6 discussed a number of methods for instrument reliability and validity assessment. These methods were employed on scales in this instrument, with findings compared to those reported in the PISA 2000 Technical Report (Adams and Wu, 2002, p. 244).

8.3.1. Reliability of Scales

The PISA 2000 Technical Report (Adams and Wu, 2002, p. 244) includes scale reliability indices for countries in which the questionnaire was administered. These

Chapter 8. Computer Familiarity Scale

findings are very useful for comparison with results from this research study, particularly since the questionnaire was administered to a different sample than was the case in the PISA study. Some items in the adapted survey were also modified (see Section 8.2), which gives further reason to re-assess the scale reliability.

As discussed in Chapter 6, Section 6.7.1, a common formula for estimating internal test reliability is Cronbach’s alpha (Alagumalai and Curtis, 2005). Reliability of the Computer Familiarity scales from the PISA 2000 study accounts for three sub-scales; “COMAB” (Comfort With and Perceived Ability to Use Computers), “COMUSE” (Computer Usage) and “COMATT” (Interest in Computers) (Adams and Wu, 2002, p. 244). Mean reliability indices for participating OECD countries were 0.83 for COMAB, 0.82 for COMUSE and 0.68 for COMATT. Table 8.9 compares scale reliability of the PISA study and the VCS research study.

Table 8.9.: Reliability Comparison for Computer Familiarity Scales

Scale	PISA Study*		VCS Study	
	# of Items	Reliability	# of Items	Reliability
COMAB	4	0.83	4	0.89
COMUSE	6	0.82	5	0.53
COMATT	4	0.68	4	0.67
COMFREQ	4	Not reported	4	0.30

* Mean reliability indices from participating OECD countries.

See Adams and Wu (2002, Table 84, p. 244) for country-specific details.

Cronbach’s alpha scores for this study were calculated using the Scale Reliability Analysis function in SPSS version 16. Please note that the table includes a fourth sub-scale, “Computer Usage Frequency” (COMFREQ), which was not accounted for in the PISA 2000 Technical Report (see Section 8.2.4 for more details). In comparison, the COMAB and COMATT scales seems to be operating on a similar level of reliability. The adapted COMUSE scale, on the other hand, did not achieve the high level of reliability as was the case with the original scale used in the PISA 2000 study. This particular scale, however, was significantly modified in this study with three original items dropped, a further two items merged and three new items

introduced. The adverse effect on this scale's reliability may have been caused by these adaptations.

The COMFREQ sub-scale, for which there is no reliability index available in the PISA 2000 Technical Report, shows the lowest level of reliability of all the scales employed in this particular instrument. Although only minor changes were made to this scale, it is nonetheless possible that these alterations affected its reliability. This cannot, however, be established since there are no means for comparison with the original scale's reliability.

If one considers a Cronbach's α of .70 or higher as representing an acceptable level, only the COMAB scale was found to be reliable in this study while COMATT comes very close.

8.3.2. Confirmatory Factor Analysis

It is possible, by studying the observable (i.e. manifest) variables, to indirectly obtain information about the latent variable (Long, 1987). Or, as Curtis (2005, p. 179) pointed out, "the constructs of interest in the social sciences are often complex and are observed indirectly through the use of a range of indicators". Confirmatory Factor Analysis, which was discussed in Section 6.7, is one method that can be employed for achieving this. CFA may be used to test a number of different factor models in order to determine what (if any) observed variables are affected by a unique factor (latent variable). Or, in other words, to "confirm that a set of variables define a construct" (Phakiti, 2007, p. 39).

Some indicators (question items) adapted from the PISA study do not, however, imply any such unobservable constructs. Specifically, the Computer Usage (COMUSE, Section 8.2.2) and Computer Usage Frequency (COMFREQ, Section 8.2.4) sub-scales were not subjected to further structure analysis for this reason.

It should be noted that the PISA 2000 Technical Report treats the COMUSE scale as a latent construct (Adams and Wu, 2002). This research study, on the other hand, is in disagreement with this argument since the question items that belong to this scale are concerned with the frequency of computer and software usage. The observed variables belonging to the scale do not suggest that they are affected by a unique factor.

8.3.2.1. Comfort With and Perceived Ability to Use Computers (COMAB/COMCOMF)

As discussed in Section 8.2.1, this particular scale consists of four question items. In the factor analysis of COMAB, the PISA 2000 Technical Report included all four items. In the current research study, however, it is suggested that the fourth question (“...how would you rate your ability to use a computer?”) should not be included in the model. While the first three questions all indicate the presence of a single latent variable of ‘computer comfort’, the fourth item suggests a separate factor of ‘perceived ability’. Further, the response categories of question four operate in a different dimension as they deviate from those of the three preceding questions in this scale. For these reasons, the single-factor model defined for this study does not include the fourth question (item code IT03Q01/ABILITY in Tables 8.1/8.2), and the latent factor has thus been renamed from COMAB to COMCOMF in this study.

The structure of the single-factor model for this item (as shown in Figure 8.3) suggests the presence of a latent variable (or factor), ‘Computer Comfort’, that may be indirectly observed through the three remaining question items.

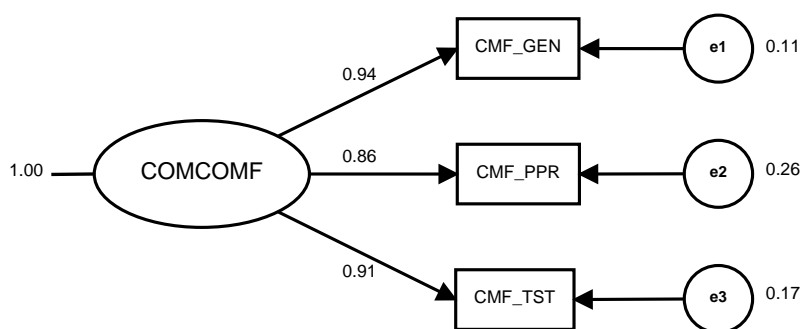


Figure 8.3.: Single-Factor Model for Computer Comfort (COMCOMF)

CFA was used to test the structure against data collected from the observed variables in order to find out if they were in fact affected by a unique factor ‘Computer Comfort’. LISREL’s report from the CFA run, as shown in Table 8.10, suggest very strong factor loadings ($>.7$).

The goodness of fit statistics were also encouraging, as Chi-Square values and the RMSEA were reported to be 0. Based on these findings, the LISREL report

Table 8.10.: Factor Loadings of the Single-Factor Model for Computer Comfort

Variable	Loadings	Standard Error
CMF_GEN	0.94	0.11
CMF_PPR	0.86	0.26
CMF_TST	0.91	0.17

N = 86

concluded that the model is saturated and that the fit is perfect. As such, LISREL did not report on other fit indices (e.g. ECVI, RMR, GFI, and CFI).

One potential issue, however, is that LISREL reports the degrees of freedom (df) as zero. Jöreskog (1969, p. 201) states that “if a value of χ^2 is obtained which is close to the number of degrees of freedom, this is an indication that the model ‘fits too well’. Such a model is not likely to remain stable in future samples and all parameters may not have real meaning.” And, as Byrne (1989) points out; it is highly unlikely to achieve a perfect fit between a hypothesised model and the observed data.

8.3.2.2. Interest in Computers (COMATT)

Information about this particular scale, including modifications made for this study, can be found in Section 8.2.3. Unlike the single-factor model discussed above, no items were dropped in the measurement model for Interest in Computers.

The structure of the single-factor model for this item (as shown in Figure 8.4) suggests the presence of a latent variable, ‘Computer Attitude’, that may be indirectly observed through the four items.

Overall factor loadings for the model, which can be observed in Table 8.11, suggest a medium to high fit for most items (Shevlin and Miles, 1998). The exception is CMP_IMP, whose factor loading is below medium and error is quite high.

Since this model did not achieve a perfect fit, LISREL provided more detailed Goodness of Fit Statistics. As shown in Table 8.12, the χ^2 value with associated degrees of freedom is non-significant ($3.86/2 = 1.93$) and the value of p is greater than the recommended threshold of .05. Although this suggests a good fit, caution must be exerted since the sample size is relatively small. Schumacker and Lomax (2004) alert the researcher to the fact that low sample size (<100) may cause the

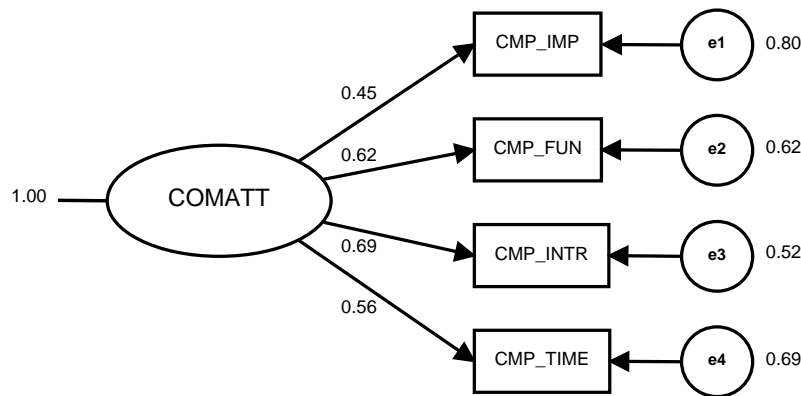


Figure 8.4.: Single-Factor Model for Interest in Computers

Table 8.11.: Factor Loadings of the Single-Factor Model for Interest in Computers

Variable	Loadings	Standard Error
CMP_IMP	0.45	0.80
CMP_FUN	0.62	0.62
CMP_INTR	0.69	0.52
CMP_TIME	0.56	0.69

N = 86

χ^2 statistic to indicate non-significant probability levels.

Looking at the RMSEA index, a poor fit (>0.1) can be observed. Further, PGFI is reported to be quite low ($\sim .50$ would be acceptable). The remaining indices (e.g. RMR, GFI, AGFI and CFI), however, reside within the ranges of what would be considered as indicative of a good fit.

Table 8.12.: Goodness of Fit Statistics for Interest in Computers

Chi-Square (χ^2)	P	df	RMSEA	RMR	GFI	AGFI	PGFI	CFI
4.12	0.127	2	0.112	0.0046	0.96	0.88	0.31	0.98

Due to the discrepancies in the goodness-of-fit statistics, the unidimensionality of the scale will be verified using Rasch model item analysis. CMP_IMP will be given careful attention due to its poor fit in the model. This is discussed in Section 8.3.3.

8.3.3. Rasch Model Item Analysis

The third and final method of scale structure analysis used in this research study is the application of the Rasch model, specifically the Rating Scale Model (RSM). Section 6.7.3 in Chapter 6 introduced this approach for measurement in more detail. The RSM was employed in order to establish scale unidimensionality through an evaluation of the precision and quality of measurement.

8.3.3.1. Comfort With and Perceived Ability to Use Computers (COMAB/COMCOMF)

Details about this scale is provided in Section 8.2.1. The last question in the scale (item code IT03Q01/ABILITY in Tables 8.1/8.2) employs different response categories than the preceding three questions. For this reason it is excluded from the analysis.

As shown in Table 8.13, the Infit Mean Square (Infit MNSQ) for item CMF_TST falls well within what is considered an acceptable range for the RSM.

Table 8.13.: Response Model Parameter Estimates for Comfort With and Perceived Ability to Use Computers (COMCOMF)

Variables	Estimates	Error	Unweighted Fit		
			<i>INFIT MNSQ</i>	<i>CI</i>	<i>t</i>
CMF_GEN	-0.231	0.244	0.54	(0.70, 1.30)	-3.6
CMF_PPR	-0.887	0.247	0.58	(0.70, 1.30)	-3.2
CMF_TST	1.118*	0.347	0.82	(0.70, 1.30)	-1.2

Separation Reliability = 0.720 **constrained*
Chi-square test of parameter equality = 13.80
df = 2
Significance Level = 0.001

The other two items fall just outside the range of 0.6 — 1.4 (suggested by Wright and Linacre (1994) as reasonable). However, their factor loadings in the CFA model were very strong and their delta values are in increasing order (which suggests that the response categories are also in order). No items in this scale were dropped.

8.3.3.2. Interest in Computers (COMATT)

Specifics about this scale, which consists of four items, can be found in Section 8.2.3. Table 8.14 shows that all items except CMP_IMP fell within the acceptable Infit MNSQ range. Moreover, CMP_IMP had a low factor loading in the CFA model as well (as was shown in Table 8.11), accompanied by a high standard error. It was therefore determined that CMP_IMP was not fit for inclusion in the path analysis. Table 8.15 shows the new estimates after removing CMP_IMP from the scale. The remaining items were still performing well and were therefore included in the final analysis.

Table 8.14.: Response Model Parameter Estimates for Interest in Computers (COMATT)

Variables	Estimates	Error	Unweighted Fit		
			INFIT	MNSQ	CI
CMP_IMP	-3.087	0.271	0.15	(0.70, 1.30)	-9.2
CMP_FUN	-0.086	0.226	1.30	(0.70, 1.30)	1.8
CMP_INTR	1.366	0.210	1.34	(0.70, 1.30)	2.1
CMP_TIME	1.807*	0.410	1.20	(0.70, 1.30)	1.3

Separation Reliability = 0.989 **constrained*
Chi-square test of parameter equality = 172.58
df = 3
Significance Level = 0.000

Table 8.15.: Response Model Parameter Estimates for Interest in Computers (COMATT)

Variables	Estimates	Error	Unweighted Fit		
			<i>INFIT</i>	<i>MNSQ</i>	<i>CI</i>
CMP_FUN	-1.057	0.220	1.39	(0.70, 1.30)	2.3
CMP_INTR	0.318	0.205	1.01	(0.70, 1.30)	0.1
CMP_TIME	0.738*	0.301	1.03	(0.70, 1.30)	0.2

Separation Reliability = 0.952 **constrained*
Chi-square test of parameter equality = 25.49
df = 2
Significance Level = 0.000

8.4. Considerations and Limitations

8.4.1. CFA and Sample Size

As discussed in the introductory chapter (Section 1.4), the number of respondents to the VCS HCI Survey was low (N=86). Phakiti (2007, p. 54) draws attention to Bollen (1990), Breckler (1990), Cudeck and Henly (1991) and Marsh et al. (1996) who all suggest that “certain goodness-of-fit indices are dependent upon sample size”. Exactly what qualifies a sample size as “sufficient”, however, is not completely agreed upon. Phakiti (2007, p. 59) lists a number of various recommendations from the literature: Boomsma (1987) recommended 400 subjects, Baldwin (1989) said at least 200, Lomax (1989) suggested at least 100 (if not 200) and Ding et al. (1995) suggested a minimum sample size of 100 to 150. Anderson and Gerbing (1984) highlights the significance of number of items per factor and suggest that a sample size of 100 is usually sufficient, providing a factor as three or more indicators. Kelloway (1998) also suggest at least two or three indicators for every latent variable. Russell (2002) concludes that CFA may produce accurate results using a sample of less than 100 cases.

Eighty-six students completed the VCS HCI survey. However, while the size of this sample might fall short of the minimum numbers recommended by some, that

sentiment is not shared by all. It has been argued, for example, that the influence of sample size depends on other design characteristics (e.g. small factor-to-variable ratio), meaning that any rules of thumb with regard to minimum sample size are invalid (Preacher and MacCallum, 2002). Similarly, in a different article, MacCallum et al. (2001, p. 613) argue that “previous studies of the sample size question in factor analysis have not explicitly taken into account the existence or influence of model error”. Finally, this research is concerned with a *population* and findings from which cannot be generalised. While CFA was undertaken and reported in this thesis, the number of respondents should nonetheless be taken into consideration as a potential limitation of this study.

8.5. Summary

The Computer Familiarity Scale, which is an adaptation of an instrument used by OECD in the PISA 2000 Study (Adams and Wu, 2002), constitutes part of the overall VCS HCI Survey. Two sub-scales of this instrument (COMCOMF and COMATT) were subjected to scale structure analysis in order to ascertain their hypothesised models’ fit the observed data. These sub-scales were also subjected to Rasch Model Item analysis, specifically the Rating Scale Model, in order to ascertain the data’s fit with the model. Findings suggests that one item in the COMATT scale, CMP_IMP, did not fit the model. As will be highlighted in Chapter 10, this item was dropped and not included in the path analysis.

Chapter 9.

User Interaction Satisfaction Scale

Quantitative facts cease to seem simply the 'given.' They must be fought for and with, and in this fight the theory with which they are to be compared proves the most potent weapon.
(Andrich, 2004, p. 113)

As described in the previous chapter, an adapted version of the Computer Familiarity Scale (Adams and Wu, 2002) sought to learn more about the population's comfort with and perceived ability to use a computer, as well as frequency and type of computer/software usage. The inclusion of this scale aimed to elicit information that could help substantiate findings from a second instrument concerned with users' reactions, attitudes and overall satisfaction with the VCS; the User Interaction Satisfaction Scale. Users may, for example, rate a software favourably on every measure of performance, yet still be dissatisfied with the system and its interface (Chin et al., 1988). The User Interaction Satisfaction Scale, which was the second instrument to form the VCS HCI Survey (see Figure 9.1), aimed at measuring the population's satisfaction with the VCS software across a range of indices. Data from the VCS HCI Survey were employed to assist in addressing Research Problem 2, as this information provides insights into software-specific constituents (e.g. input/output (I/O), graphical user interface (GUI), stability, speed, usability, extensibility, reliability and functionality) from the end-users' perspective.

It should be pointed out that this part of the VCS HCI Survey elicits the 'computer science' aspects of the VCS research study. It provides important feedback directly

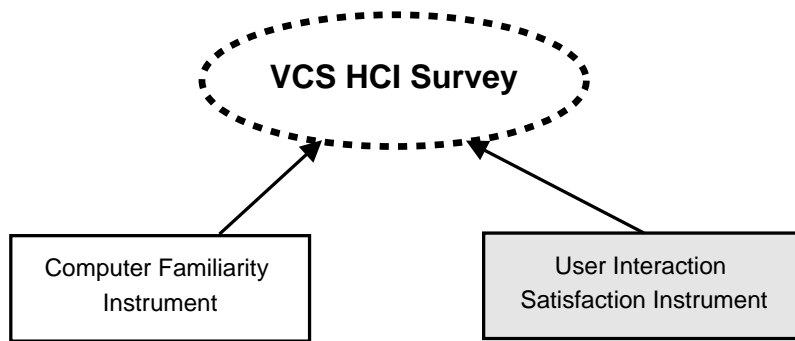


Figure 9.1.: VCS HCI Survey Composition

from the target user group about the usability of the software, which may reflect back on software implementation and design. However, the instrument also provides information that is crucial for determining the VCS' value as a training tool for education. Thus, the trans-disciplinary nature of this study, which has already been emphasised a number of times in this document, is also highly relevant in the design and analysis of this particular instrument.

The focus of this chapter, which is structurally similar to the previous, concerns the user interaction satisfaction scale, adaptations made and analysis of its structure.

9.1. Data

This instrument was administered in the same survey as the Computer Familiarity Scale (e.g. the VCS HCI Survey) and shares the same considerations as described in Chapter 8, Section 8.1.

9.2. The Instrument

In his book on designing the user interface, Shneiderman (1992) included a short and a long version of the Questionnaire for User Interaction Satisfaction (QUIS); a measurement tool designed by a multi-disciplinary team of researchers to assess the users' subjective rating of specific aspects of a human-computer interface (Harper and Norman, 1993). The questionnaire, which has been continuously refined and updated for more than two decades, comprises a number of scales that

can be generically configured (e.g. ‘mixed and matched’) according to the needs and interests of the interface analysis. The measure is labelled by the developers as “highly reliable across many types of interfaces” (Harper and Norman, 1993, p. 224), although the claim may not hold for user-customised versions (Norman, 2009).

The most recent edition of the QUIS (version 7) is “arranged in a hierarchical format and contains: (1) a demographic questionnaire, (2) six scales that measure overall reaction ratings of the system, (3) four measures of specific interface factors: screen factors, terminology and system feedback, learning factors, system capabilities, and (4) optional sections to evaluate specific components of the system: technical manuals and on-line help, on-line tutorials, multimedia, Internet access and software installation” (Harper et al., 1997, p. 2). A number of items and scales in this instrument were, however, not applicable for use in this study. A shorter form of the QUIS (v. 5.0), published by Chin et al. (1988), was therefore deemed better suitable as it included the main elements needed for the context of this research. A capture of the original instrument can be found in Figure 6.8 on page 150. It is comprised of five scales; “Overall User Reactions”, “Screen”, “Terminology and System Information”, “Learning” and “System Capabilities”. All five scales were used in the current study although some items were modified or excluded. Item-specific adaptations made to the questionnaire are described in more detail later in this chapter.

As established in Chapter 8, the Computer Familiarity Scale employed four response categories for its Likert-type scales. The QUIS version considered for this study, on the other hand, used a 10-point scale for its items (0-9). Each QUIS item has associated with it a negative adjective on the left and a positive adjective on the right (e.g. verbal response descriptors). The adapted version used in this study kept the original labelling, but the 10-point scale was reduced to 4 (1-4) plus an N/A option for all items. Section 9.3 will cover the reduction of response categories in more detail.

The adapted version of the QUIS and the Computer Familiarity Scale were both administered through the on-line VCS HCI Survey. A separate page was reserved for each of the five QUIS instrument scales (along with the qualitative component “User’s Comments”); a layout recommended by the QUIS developers (Harper and

Norman, 1993). In a comparison between on-line and paper-based formats of the QUIS, no difference was found in the users' ratings (Slaughter et al., 1994). It was, however, found that subjects using the on-line version provided clearer and more substantiated feedback in the comments sections than those using the paper-based version (Slaughter et al., 1994). Screen-shots of the final representation and layout of the questionnaire used in this study are available in Appendix S. It may also be useful to refer to the codebook for this instrument, which can be found in Appendix T.

The following subsections describe each sub-scale (shown in Figure 9.2) in more detail and will also, where applicable, highlight what items were altered or omitted from the original scale.

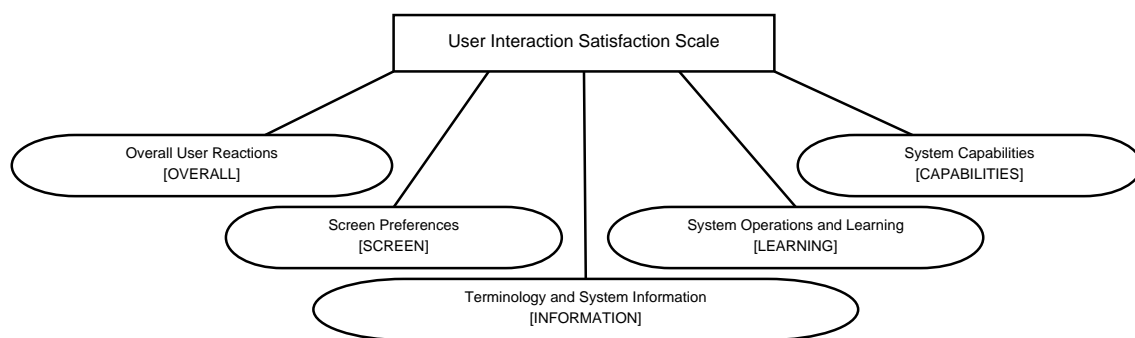


Figure 9.2.: Structure of the User Interaction Satisfaction Scale

9.2.1. Overall User Reactions Scale (VCSREA)

This sub-scale is comprised of six items involving the user's first impression of the VCS software, which is reflected by the items' adjective labels (see Table 9.1). The scale is very useful for eliciting information about the VCS' perceived "face value"; an aspect that may be indicative of a) further, more in-depth, feedback about the VCS, b) overall software design and c) added/reduced motivation for further use of the tool. As will be described later in this chapter, the scale underwent both structural and item analysis to establish the existence of any latent measures as well as its overall validity and reliability.

Aside from the abovementioned reduction to a 4-point scale (which applies to all sub-scales), no changes were made to the items of this scale. An introductory text

was however added to the scale: “Please select the numbers which most appropriately reflect your impressions about using the Virtual Classroom Simulation. Not applicable = N/A.”

There was one question text common for the six items of this scale, which originally read “Overall reactions to the software:”. In the adapted version, the word ‘software’ was changed to ‘Virtual Classroom Simulation’. Table 9.1 lists the adjective labels for the six items in the scale “Overall User Reactions”.

Table 9.1.: Overall User Reactions Sub-Scale

<i>Overall Reactions to the Virtual Classroom Simulation:</i>		
Item	Negative Label	Positive Label
1	Terrible	Wonderful
2	Frustrating	Satisfying
3	Dull	Stimulating
4	Difficult	Easy
5	Inadequate power	Adequate power
6	Rigid	Flexible

A screenshot of this scale’s layout in the VCS HCI Survey is available in Appendix S, Figure S.4.

9.2.2. Screen Preferences Scale

Two of the four items in the original QUIS sub-scale were not included in the adapted version. Specifically, it was found that the items “*Highlighting on the screen simplifies task [not at all/very much]*” and “*Sequence of screens [confusing/very clear]*” were not useful in this study.

Table 9.2.: Screen Sub-Scale

Item	Negative Label	Positive Label
In terms of readability, how did you find the text on the screen?	Hard to read	Easy to read
Were the screen layouts helpful?	Not at all	Very helpful

The first item in Table 9.2 was reworded from original text “*Characters on the computer screen*”, but the labels were kept as per the original scale. The second

item was changed from “*Organization of information on screen*” to “*Were the screen layouts helpful?*”, a change inspired by a more recent version of QUIS (Shneiderman, 1992, p. 486).

A screenshot of this scale’s layout in the VCS HCI Survey is available in Appendix S, Figure S.5.

9.2.3. Terminology and System Information Scale

The original sub-scale consisted of six items in total. One of the items, “*Use of terms throughout the system [inconsistent/consistent]*”, was not included in the adapted version. The second item, “*Computer terminology is related to the task you are doing [never/always]*”, was altered to read “*Does the terminology relate well to your education course*”. Items three and four, which originally read “*Position of messages on screen*” and “*Messages on screen which prompt user for input*”, were merged and changed to “*How would you rate the help, information and error messages that appeared in the program?*”. The item “*Computer keeps you informed about what it is doing*” was modified to “*Did the program keep you informed about what it was doing?*”. Finally, the last item was rewritten from “*Error messages*” to “*Did you get any error messages? If so, how helpful were they?*”.

Table 9.3 shows the altered versions of the items in this scale along with the negative and positive labels (the majority of which were kept as per the original scale).

Table 9.3.: Terminology and System Information Sub-Scale

Item	Negative Label	Positive Label
Does the terminology relate well to your education course?	Unrelated	Well related
How would you rate the help, information and error messages that appeared in the program?	Inconsistent	Consistent
	Confusing	Clear
Did the program keep you informed about what it was doing?	Never	Always
Did you get any error messages? If so, how helpful were they?	Unhelpful	Helpful

A screenshot of this scale's layout in the VCS HCI Survey is available in Appendix S, Figure S.6.

9.2.4. System Operations and Learning Scale

Of the original six items for this sub-scale, only three were relevant to this study. Hence, the items "*Exploration of features by trial and error*", "*Remembering names and use of commands*" and "*Supplemental reference materials*" were not included in the adapted version of the scale.

"*Learning to operate the system*" was altered to "*How difficult was it to learn to use the VCS*", "*Tasks can be performed in a straight-forward manner?*" to "*Could the tasks in the program be performed in a straight-forward manner?*" and "*Help messages on the screen [unhelpful/helpful]*" was changed to "*How would you rate the help messages in the program?*".

Table 9.4 lists the items in the adapted scale along with the adjective labels which were kept identical to the original scale.

Table 9.4.: Learning Sub-Scale

Item	Negative Label	Positive Label
How difficult was it to learn to use the VCS?	Difficult	Easy
Could the tasks in the program be performed in a straight-forward manner?	Never	Always
How would you rate the help messages in the program?	Confusing	Clear

A screenshot of this scale's layout in the VCS HCI Survey is available in Appendix S, Figure S.7.

9.2.5. System Capabilities Scale

The fifth and final sub-scale in the short version of the QUIS comprised a total of 5 items. Similar to a few of the other scales, some items were not included in the adapted version, as they were incompatible with the VCS software and/or did not

relate to the purpose of this study. The items “*System tends to be*” and “*Correcting your mistakes*” were therefore excluded from the adapted scale.

“*System speed*” was altered to read “*How was the speed of the VCS?*”, “*System reliability*” was changed to “*How reliable was the program?*” and the adapted version of the item “*Experienced and inexperienced users’ needs are taken into consideration*” read “*Do you think the needs for both experienced and inexperienced users have been taken into consideration?*”.

The three items that were maintained in the adapted scale, along with their associated adjective labels (as per the original scale), are listed in Table 9.5.

Table 9.5.: System Capabilities Sub-Scale

Item	Negative Label	Positive Label
How was the speed of the VCS?	Too slow	Fast enough
How reliable was the program?	Very unreliable	Very reliable
Do you think the needs for both experienced and inexperienced users have been taken into consideration?	Never	Always

A screenshot of this scale’s layout in the VCS HCI Survey is available in Appendix S, Figure S.8.

9.2.6. Comments (Qualitative)

The final screen of the VCS HCI Survey included an optional comments section. While this item was not present in the QUIS version adapted by this study (Chin et al., 1988), it can be observed in the version published by Shneiderman (1992, p. 487). The question sought to elicit general comments from the students with regard to the VCS, trial or the survey. Data collected from the question were only used for information purposes and not analysed any further.

A screenshot of this section’s layout in the VCS HCI Survey is available in Appendix S, Figure S.9.

9.3. Reduction of Item Response Options

According to Chang (1994), a number of researchers have tried to find the ‘optimal’ number of scale points that maximises the reliability of the summative rating scale. Studies on this rating scale, perhaps better known as the “Likert-type scale” (Likert, 1932), are contradictory Chang (1994) says. To demonstrate this, he points out a number of publications claiming that reliability is independent of the number of scale points as well as researchers who maintain that 7, 5, 4 or 3-point scales yield the highest level of reliability. When looking at textbooks on the subject, Dawes (2008) found that 5- or 7-point formats are the most commonly used although Jacoby and Matell (1971, p. 498) provide evidence to suggest that “both reliability and validity are independent of the number of scale points used for Likert-type items”. As such, they argue that neither reliability nor validity should influence the rating format of the Likert scale (Matell and Jacoby, 1971). Jamieson (2004), on the other hand, is not as concerned with the number of response categories as she is with *how* the scales are treated in the analysis of data.

The User Interaction Satisfaction Scale employed in this study is an adaptation of QUIS version 5.0. The original version, as presented by Chin et al. (1988, p. 215), uses a 10-point Likert-type scale (0-9) and does not have an ‘N/A’ option. The developers argued that “a questionnaire’s reliability is related to the number of items and scaling steps” and that “10 point scales were used since more than 10 steps would contribute little to reliability” (Chin et al., 1988, p. 214). In subsequent versions of QUIS the number of point scales was reduced from 10 to 9 (1-9) along with the addition of an ‘N/A’ option (e.g. Shneiderman, 1992), though the developers did not justify these Likert-related alterations any further.

The 10-point Likert-type scale used in QUIS 5.0 (Chin et al., 1988) was, in this research study, reduced to 4 (1-4). A “not applicable” (‘NA’) option was also added to each item. HCI VCS Survey consistency with the Computer Familiarity Scale is one argument for the reduction of item response options. Another, more important, factor for doing so was to eliminate the mid-point of the Likert-type scale so as to control for bias; e.g. situations where respondents have a desire “to please the interviewer or appear helpful or not be seen to give what they perceive to be a socially unacceptable answer” (Garland, 1991, p. 1). Garland (1991) and Bartlett

et al. (1960) provide some evidence to suggest that the removal of the mid-point, whereby ‘forcing’ respondents away from a neutral opinion, minimises and provides better control over social desirability/attitude bias. While a scale without a mid-point does not change the direction of opinion, Garland (1991) says, the intensity of opinions may, however, be altered. The direction (e.g. negative or positive end of the scale) in which respondents are pushed when the neutral mid-point is absent appears to be content specific; Garland (1991) found in his research that respondents were pushed to the negative end of the scale, whereas Worcester and Burns (1975) found the opposite.

In response to the contradictory nature of Likert-type scale research findings, the adapted version of the QUIS used in this study underwent instrument reliability and validity analysis which was compared with findings from the developers’ own analysis of QUIS.

9.4. Scale Structure Analysis

The adapted version of the QUIS underwent a structure analysis process similar to that of the Computer Familiarity instrument. However, the QUIS adaptation used in this study comprised more items and scales, underwent more substantial changes and had not previously been subjected to the same level of rigour with regard to reliability and validity analysis as was the case with the PISA-based instrument. The opportunities (and necessity!) for validity and reliability analysis were therefore emphasised.

9.4.1. Reliability of Scales

Some reliability data are available for the original QUIS scale, e.g. Chin et al. (1988, QUIS v. 5.0). This section will provide a full account for reliability findings in this study and compare these with the findings for QUIS v. 5.0.

Section 9.3 argued for the reduction of item response options in the Likert-type scale, as well as the elimination of its mid-point. Further to these changes, the instrument also underwent scale and item-specific alterations as described in Section 9.2. These factors should be taken into consideration in the assessment

and comparison of the reliability of the original and adapted versions of the scales. The Cronbach's α formula was employed in the estimation of internal test reliability, using the same techniques as for the Computer Familiarity Scale (see Chapter 8, Section 8.3.1). While Chin et al. (1988) provided only the overall reliability for their instrument, this study does also provide reliability indices for each sub-scale (Table 9.6).

Table 9.6.: Reliability Comparison for User Interaction Satisfaction/QUIS Sub-Scales

NOTE:
This table is included on page 247 of the print copy of
the thesis held in the University of Adelaide Library.

9.4.2. Confirmatory Factor Analysis

Papers published by members of the team behind the QUIS imply the involvement of factor analysis in order to establish the level of construct validity. Chin et al. (1988) subjected version 5.0 of the instrument to EFA in an effort to determine item commonalities. This exercise produced four latent factors that were labelled 1) Learning, 2) Terminology and Information Flow, 3) System Output and 4) System Characteristics (Chin et al., 1988, p. 216). A subsequent paper on QUIS (version 5.5) by Harper and Norman (1993, p. 224) also suggest the presence of four "specific interface factors", which they named 1) Screen Factors, 2) Terminology and System Feedback, 3) Learning Factors and 4) System Capabilities. Here, Harper and Norman (1993, p. 228) explicitly state that "the four separate interface factors measured in the QUIS are hypothesized to explain unique facets of total satisfaction". Aside from the different labelling of factors, as shown in Table 9.7, there are very few item-specific differences between QUIS

version 5.0 and 5.5.

Table 9.7.: QUIS Interface Factors

Chin et al. (1988)	Harper and Norman (1993)
Learning	Learning Factors
Terminology and Information Flow	Terminology and System Feedback
System Output	Screen Factors
System Characteristics	System Capabilities

Recall that the short form of the QUIS also included a fifth sub-scale (labelled in this study as “Overall Reactions to the VCS”, see Section 9.2.1). Curiously, the developers did not subject this scale, which they named “Overall Reactions to the Software”(Chin et al., 1988)/“Overall System Satisfaction”(Harper and Norman, 1993), to factor analysis akin to what was the case for the other four scales. The word ‘curiously’ is used here because this particular measure of *overall reactions/satisfaction* to a system is arguably more indicative of a latent construct than the items that led to the establishment of the four factors listed in Table 9.7. This research study therefore makes two arguments that might conflict with the proposals of the QUIS developers.

First, while the sub-scale of *overall reactions/satisfaction* was not subjected to factor analysis by the QUIS team, this research study argues that it should have been. The six items of which this scale is constituted (see Table 9.1 on page 241) involve affectionate/attitudinal connotations that, combined, seek to discover the user’s *overall* reactions to the software. *Satisfaction*, *reactions* and *attitudes* are affective terms that Sexton and Klinec (2001) use to represent their latent traits. Likewise, Morgan and Casper (2000) examined the factor structure (EFA and CFA) of participant *reactions* to training.

Second, this study does not see a rationale for performing EFA/CFA on what the QUIS developers hypothesised to be four separate interface factors. One could argue that the items that, according to the QUIS developers, make up the four latent constructs can all be individually observed and tend to lean more towards factual rather than affective statements. Recall that EFA and CFA are used to measure constructs that are not easily observed (Diamantopoulos and Sigauw, 2000). In this case, it would be fair to say that the observed variables that Chin et al. (1988)

and Harper and Norman (1993) claim are measuring their four latent factors are not actually doing so. For example, it is difficult to see a latent connection between, say, “System tends to be [noisy/quiet]” and “Correcting your mistakes [difficult/easy]”.

For the reasons given above, CFA will in this research study be performed only on the sub-scale of “Overall Reactions to the VCS”.

9.4.2.1. Overall User Reactions Sub-Scale (VCSREA)

The hypothesised model for this scale, as shown in Figure 9.3, suggests an underlying factor which indicates the overall reactions to the VCS.

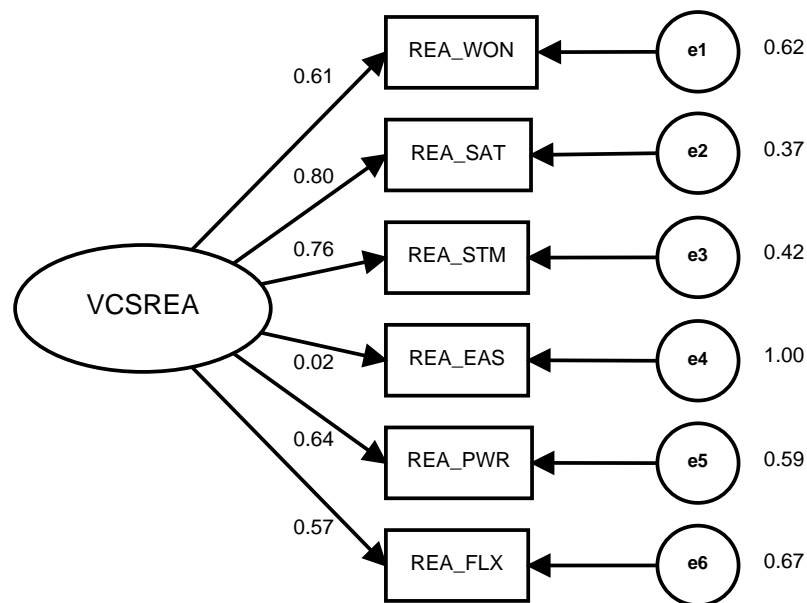


Figure 9.3.: Structure of the Single-Factor Model for the Overall Reactions to the VCS Sub-Scale

CFA, using LISREL, was used to test this structure against observed data. The factor loadings for the model, which can be observed in Table 9.8, suggest a medium to high fit for most items.

One item of concern is REA_EAS (“Overall Reactions to the VCS”, “Difficult/Easy”). The factor loading for the item is extremely low with a very high standard error. It would not, however, be appropriate to remove this item from the sub-scale without further investigation. Observation of the raw data

Table 9.8.: Factor Loadings of the Single-Factor Model for Overall Reactions to the VCS

Variable	Loadings	Standard Error
REA_WON	0.61	0.62
REA_SAT	0.80	0.37
REA_STM	0.76	0.42
REA_EAS	0.02	1.00
REA_PWR	0.64	0.59
REA_FLX	0.57	0.67

N = 86

reveals that a number of respondents selected the ‘NA’ option for this item while still rating other items on the 1-4 scale. To the researcher, this seems peculiar because on face value it appears to be an easy item to rate. A likely explanation could be inherent in the design of the questionnaire layout.

As shown in Figure 9.4, the ‘N/A’ option is positioned on the positive end of the Likert scale. Aside from the ‘NA’ label at the top, there are no other positional or visual indicators that distinguish this response option as separate from the 1-4 scale. It is reasonable to believe that a number of respondents inadvertently ticked ‘NA’ while in fact they intended to select the right-most value on the scale. One student alerted to this fact in the comments section:

This is about the survey - not VCS - Maybe don't put NA on the right hand side, people might click N/A when they mean to give it the highest possible score for that question. (ST0014)

Based on this assumption the item will not be dropped from the sub-scale as of yet, but rather placed under further scrutiny in the Rasch Model Item Analysis (in Section 9.4.3).

Recall from Chapter 6 that the degrees of freedom (df) serve as a standard by which to judge whether χ^2 is large or small, and that a good fit is implied where $\chi^2/df \leq 3.0$. As shown in Table 9.9, the χ^2 value with associated degrees of freedom (i.e. $20.21/9 = 2.24$), indicates a good fit.

The value of p, however, is below the recommended threshold of .05. Although the chi-square finding implies a good fit, the sample size is relatively small. Looking

Overall reactions to the Virtual Classroom Simulation:						
	1	2	3	4	NA	
Terrible	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Wonderful
Frustrating	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Satisfying
Dull	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Stimulating
Difficult	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Easy
Inadequate Power	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Adequate Power
Rigid	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Flexible

Figure 9.4.: Survey Layout of VCSREA Sub-Scale

Table 9.9.: Goodness of Fit Statistics for Interest in Computers

Chi-Square (χ^2)	P	df	RMSEA	RMR	GFI	AGFI	PGFI	CFI
20.21	0.016	9	0.121	0.50	0.93	0.83	0.40	0.93

at the RMSEA index, a poor fit (>0.1) can be observed, although PGFI is close to what would be considered acceptable ($\sim.50$). RMR is very high (should be below 0.05 for good fit), but the remaining indices (e.g. GFI, AGFI and CFI) reside within the ranges of what would be considered as indicative of a good fit. Due to the discrepancies in the goodness-of-fit statistics, the unidimensionality of the scale will be verified using Rasch model item analysis. This is discussed in Section 8.3.3.

9.4.2.2. Screen Sub-Scale

Diamantopoulos and Sigauw (2000) caution researchers to include too many manifest variables in a single construct, as an exceedingly complex model increases the likelihood of encountering problems with the model’s fit. Further to this, the sample size requirements may also increase if a model includes too many observed variables. While there is no hard rule on the maximum number of manifest variables per latent variable, Diamantopoulos and Sigauw (2000) suggest that each latent variable should be measured by 3-4 indicators.

With regard to the Screen sub-scale adapted from the QUIS, the opposite problem occurs; the model is too simple. While the original scale comprised four items, the adapted version kept only two of these. If the conceptual model is too simple, as is the case for this particular scale (see Figure 9.5), it may affect the quality of the

measurement (Diamantopoulos and Siguaaw, 2000). This is a sentiment supported by Russell (2002) who recommends that a factor should have at least three observed variables.

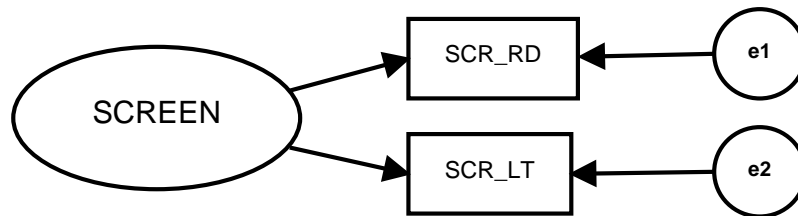


Figure 9.5.: Structure of the Single-Factor Model for the Screen Sub-Scale

Moreover, when analysing the two remaining items of the Screen sub-scale, it becomes apparent that they do not actually serve as appropriate indicators for any latent variables. Considering the above considerations, CFA was not performed on this scale.

9.4.2.3. Terminology and System Information Sub-Scale

Figure 9.6 displays what would represent a hypothesised model for the Terminology and System Information scale. However, as was the case with the Screen sub-scale, the items in this model do not form a combined set that is indicative of an underlying latent trait.

The items do, however, appear to have commonalities that justify their adherence to a shared category. Chin et al. (1988) established this category with the aid of EFA, which may well be an appropriate method for establishing item commonalities. This study does not, however, agree with the terminology used by Chin et al. (1988) and Harper and Norman (1993) which indicates that the sub-scale is measuring something that is not readily observable.

9.4.2.4. Learning Sub-Scale

The same arguments apply for this scale (visualised in Figure 9.7) as was the case for the Terminology and System Information sub-scale (Section 9.4.2.3). Thus, CFA was not carried through for this scale.

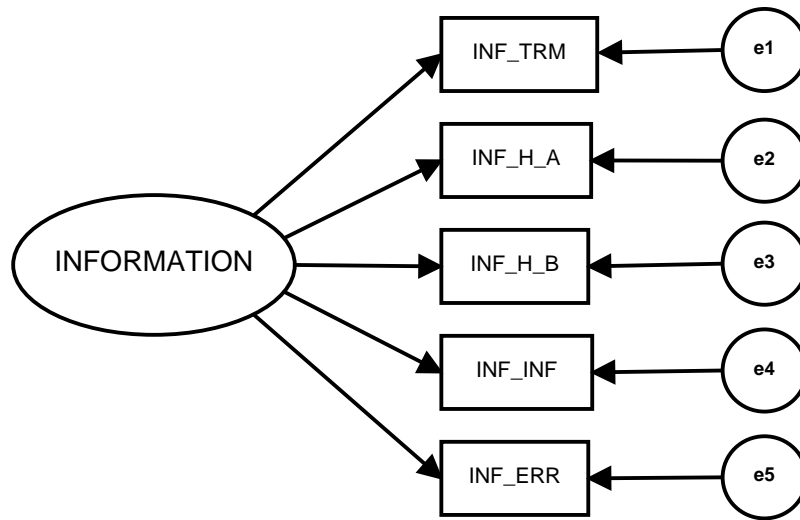


Figure 9.6.: Structure of the Single-Factor Model for the Terminology and System Information Sub-Scale

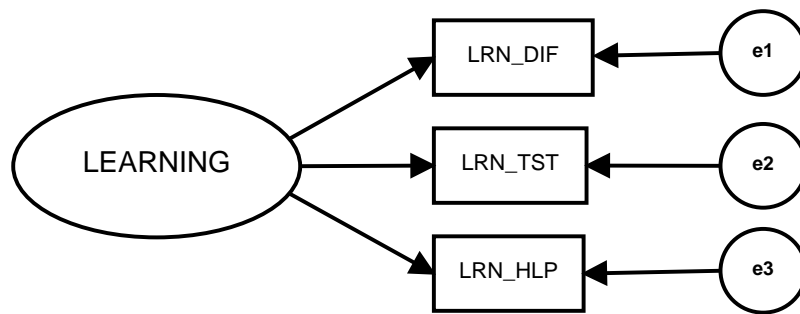


Figure 9.7.: Structure of the Single-Factor Model for the Learning Sub-Scale

9.4.2.5. System Capabilities Sub-Scale

The same arguments apply for this scale (visualised in Figure 9.8) as was the case for the Terminology and System Information sub-scale (Section 9.4.2.3). Thus, CFA was not carried through for this scale.

9.4.3. Rasch Model Item Analysis

One sub-scale from the User Interaction Satisfaction Scale qualified for Rasch Model Item Analysis; Overall User Reactions (VCSREA). The Rating Scale Model (RSM) was employed in a similar manner to what was the case for two sub-scales

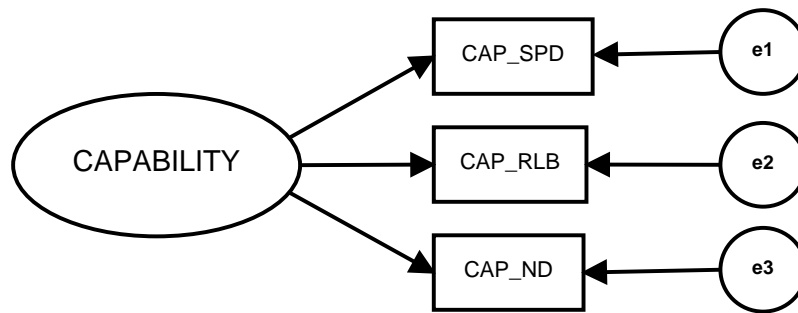


Figure 9.8.: Structure of the Single-Factor Model for the System Capabilities Sub-Scale

in the Computer Familiarity Scale (see Section 8.3.3, Chapter 8). Findings may be observed in the section below.

9.4.3.1. Overall User Reactions (VCSREA)

This sub-scale consists of five items, details of which can be found in Section 9.2.1. Table 9.10 shows that all items except REA_EAS fell within the acceptable Infit MNSQ range of 0.6 — 1.4 (suggested by Wright and Linacre (1994) as reasonable). This finding, coupled with REA_EAS' poor factor loading and a high error in the CFA, suggests that the item is not fitting and should not be included in the final path analysis.

Table 9.11 shows the new parameter estimates calculated after removing the item REA_EAS from the scale. The remaining items were still performing well and were therefore included in the final path analysis.

9.5. Considerations and Limitations

9.5.1. CFA and Sample Size

This instrument was administered in the same survey as the Computer Familiarity Scale and shares the same considerations as described in Chapter 8, Section 8.4.1.

Table 9.10.: Response Model Parameter Estimates for Overall User Reactions (VCSREA)

Variables	Estimates	Error	Unweighted Fit		
			INFIT	MNSQ	CI
REA_WON	-0.181	0.086	0.64	(0.70, 1.30)	-2.7
REA_SAT	-0.218	0.087	0.70	(0.70, 1.30)	-2.1
REA_STM	-0.075	0.085	0.82	(0.70, 1.30)	-1.2
REA_EAS	-0.479	0.089	2.24	(0.70, 1.30)	6.1
REA_PWR	0.489	0.079	1.05	(0.70, 1.30)	0.4
REA_FLX	0.464*	0.191	0.79	(0.70, 1.30)	-1.5

Separation Reliability = 0.943 **constrained*
Chi-square test of parameter equality = 78.24
df = 5
Significance Level = 0.000

Table 9.11.: Response Model Parameter Estimates for Overall User Reactions (VCSREA)

Variables	Estimates	Error	Unweighted Fit		
			INFIT	MNSQ	CI
REA_WON	-0.404	0.104	0.85	(0.70, 1.30)	-1.0
REA_SAT	-0.457	0.104	0.83	(0.70, 1.30)	-1.1
REA_STM	-0.251	0.103	1.03	(0.70, 1.30)	0.3
REA_PWR	0.575	0.096	1.29	(0.70, 1.30)	1.8
REA_FLX	0.538*	0.204	1.11	(0.70, 1.30)	0.8

Separation Reliability = 0.955 **constrained*
Chi-square test of parameter equality = 76.23
df = 4
Significance Level = 0.000

9.6. Summary

The User Interaction Satisfaction Scale, which is an adaptation of an instrument advanced by Chin et al. (1988), makes part of the overall VCS HCI Survey. One sub-scale of this instrument (VCSREA) was subjected to scale structure analysis in order to ascertain its hypothesised models' fit the observed data. The sub-scale was also subjected to Rasch Model Item analysis, specifically the Rating Scale Model, in order to ascertain the data's fit with the model. Findings suggests that one item in the scale, REA_EAS, did not fit the model. This is likely due to a flaw in the survey layout design, which positioned the 'N/A' option too close to the Likert scale values. As will be highlighted in Chapter 10, this item will not be included in the path analysis.

Chapter 10.

Improving Education Students' Understanding of Classroom Interactions

*VCS has the potential for the
education, exploration and evaluation
of different teaching methodologies*

EDUCATION STUDENT 0005 IN SURVEY RESPONSE

The next chapter will summarise and articulate how the modelling, design and development efforts of this research responded to the first two problems of this study. It will also demonstrate how these responses were further validated by research design output. Research Problem 3, however, cannot be addressed, not even in part, without involvement from education students and methods advanced by the research design:

Problem 3. To what extent is the VCS useful in reshaping preservice teachers' mental model about teaching and learning, as well as advancing their pedagogical content knowledge?

- How can this be accurately measured?
- How can the impact of the VCS be further improved?

The research design of this study has implicitly proposed that the usefulness of the VCS can be measured with the aid of exploratory, explanatory and confirmatory processes. It picks up on the qualitative research methodology and the outputs borrows from the mixed-methods designs advanced by Creswell (2008, e.g. Figure 10.1) to complement and inform the quantitative methods adopted in this research.

NOTE:
This figure is included on page 258 of the print copy of
the thesis held in the University of Adelaide Library.

Figure 10.1.: Types of Mixed Methods Designs (adapted from Creswell, 2008, p. 557)

This chapter seeks to expound on the problem of measuring VCS usefulness by confirming the theoretical model advanced in Chapters 1 and 2. This will be done by triangulating what students indicated through their qualitative feedback, and what was evidenced through data obtained through the quantitative survey questionnaire. Moreover, a sample analysis of the VCS Assignment (introduced in Section 6.3.6, Chapter 6) will be examined as it brings the viability and functionality of the VCS to its core purpose. Inferences of the findings will be made in the next chapter.

10.1. Structural Equation Modelling and Causal Relationships

Section 6.7.4 in Chapter 6 provided an overview of Path Analysis (PA), an extension of Structural Equation Modelling (SEM), and how this statistical technique may be used to examine causal relationships between variables (Jöreskog, 1994). The importance of establishing scale and item validity and reliability prior to examining these relationships was also emphasised.

10.1.1. Reliability and Validity of Variables

In Chapters 8 and 9, this was brought about for the VCS HCI Survey data by utilising confirmatory and explanatory models. Similarly for the VCS Education Survey (Chapter 7), reliability of the data quantification process was carefully managed and examined. In examining the reliability and validity of the constructs COMATT and VCSREA, it was found that two items did not fit their models. This has implications for the theoretical model that was originally postulated, because it now contains items that could distort the relationship estimates as well as the fit of the model. The theoretical model must therefore be modified to reflect these recent discoveries. Figure 10.2 shows the final theoretical model, where the rejected items (CMP_IMP and REA_EAS) are highlighted to indicate that they are no longer part of the model.

10.1.2. W-Scale and Latent Variable Scores (LVS)

Three latent constructs were identified in the the VCS HCI Survey; COMATT, COMCOMF and VCSREA. As mentioned earlier in this chapter, the Rasch model of measurement was undertaken for these constructs to identify items with a poor fit. Two items were rejected; CMP_IMP and REA_EAS. In the path analysis, the ‘surviving items’ (those not rejected) were ‘collapsed’ into latent variables determined by the factors they belong to, e.g. COMATT, COMCOMF and VCSREA (e.g. the factor COMATT is composed of the observed CMP_FUN, CMP_INTR and CMP_TIME).

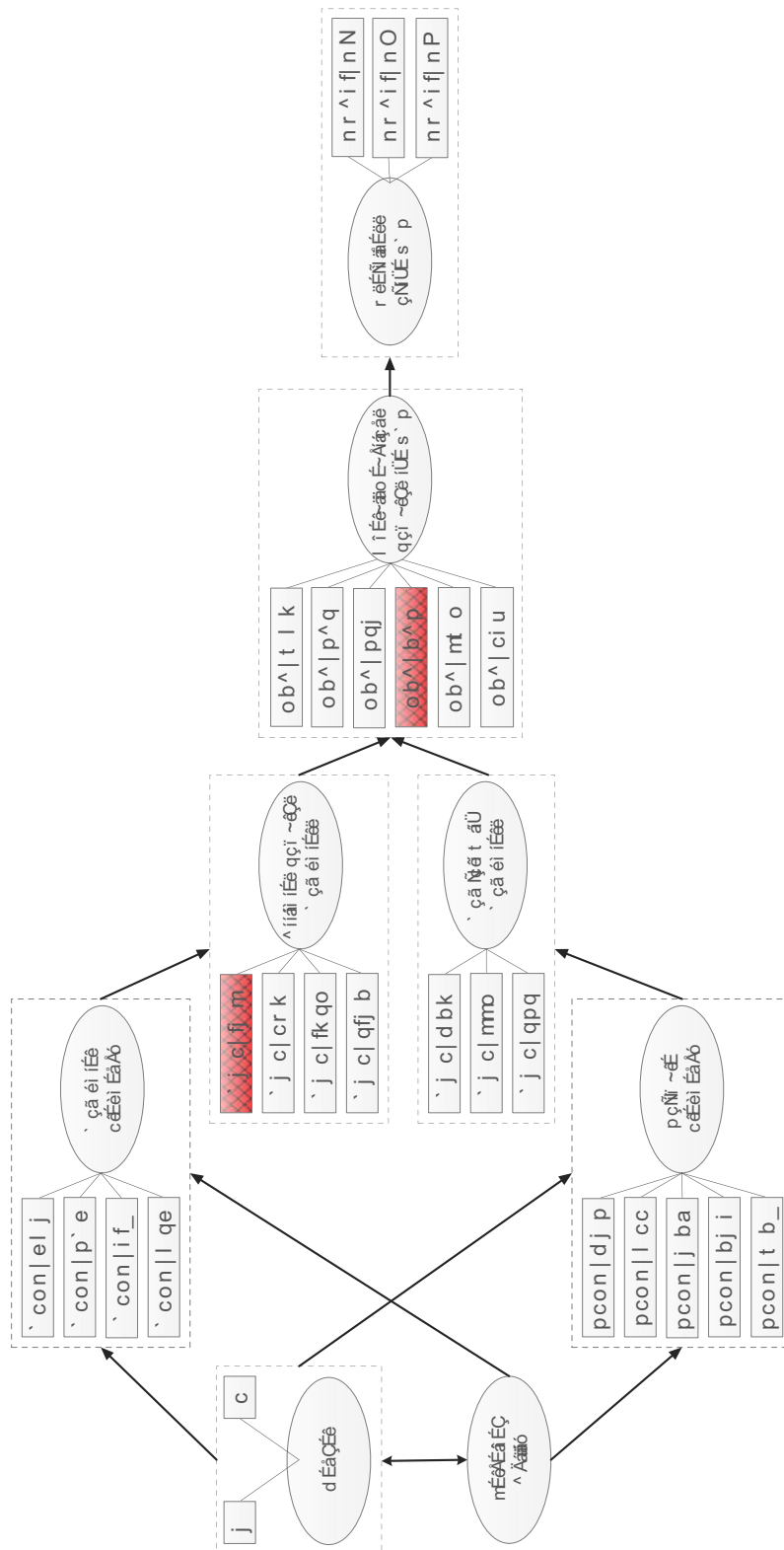


Figure 10.2.: Detailed View of the Final Theoretical Model for Analysis

Rasch ability estimates obtained for those factors that underwent Rasch analysis were subjected to W-scale transformation. This technique is of benefit as it eliminates negative and decimal values in the data, and the properties of the W-scale makes it a more convenient alternative to the logit scale (Woodcock, 1999, p. 111).

10.1.3. Missing Values

Missing values were observed when data from the VCS Education Survey (N=159) and the VCS HCI Survey (N=86) were combined. The need to ‘complete’ the data was therefore needed prior to undertaken the PA. Imputation, the practice of replacing data that is missing, “has long been recognized as an attractive approach to analysing incomplete data” (Schafer, 1999, p. 3). Multiple Imputation (MI) is a technique that “replaces each missing or deficient value with two or more acceptable values representing a distribution of possibilities” (Rubin, 2004, p. 3). MI is a preferred alternative to simply deleting cases with missing values, because the latter may cause serious bias in the estimates (Schafer, 1999; Kunnan, 1998) as well as resulting in an unacceptable loss of detail (Schafer, 1999). MI was employed using LISREL to develop the covariance matrix.

The next section will report on the path analysis outcomes, and provide a discussion on the important predictors for the usability and usefulness of the VCS.

10.2. Path Analysis

PA was used in this study be used for the purpose of investigating the possible relationships between variables in the theoretical model. The theoretical model assumes variables from the VCS HCI Survey as well as quantified data from the VCS Education Survey. Of particular interest are the variables directly related to the reactions towards the VCS, both in terms of usability (VCSREA) and its usefulness in education (QUALI_Q1, QUALI_Q2 and QUALI_Q3). Table 10.1 informs of the question text for each variable.

Qualitative responses from the VCS Education Survey provide a lot of information about students’ opinions about the usefulness of the VCS. Likewise,

Table 10.1.: VCS-Specific Variables in the Theoretical Model

Variable	Question
VCSREA	“Overall Reactions to the VCS”, as rated by 5 sub-items: (Wonderful Satisfying Stimulating Powerful Flexible)
QUALI_Q1	“How do you view the educational value of the VCS?”
QUALI_Q2	“Do you believe the VCS is an effective means of training?”
QUALI_Q3	“Do you believe the VCS generates a valuable learning experience?”

the quantified interpretation of these responses can with ease be used to provide a statistical summary of the population's view (e.g. with descriptive statistics). Question responses do not, however, give any clues as to *why* a student has a particular view of the usability and the usefulness of the VCS. A number of variables that may have an impact on the students' views about VCS were therefore included in the theoretical model. PA is useful to give information about the relative importance of these variables as *predictors* of the VCS-specific variables (Jöreskog and Sörbom, 2001). Assumptions of PA include that the direction of causation can be identified, data are at the interval level, relations are linear, data meet criteria for regression analysis and that parsimony of model is fair (Morrison, 2009, p. 98).

The confirmed path diagram is presented in Figure 10.3. This is a recursive model (Jöreskog and Sörbom, 2001; Kelloway, 1998), which is signified by its unidirectional nature meaning it can only be read in one direction, in this case left to right.

In practice, recursive regression can test any independent factor and/or dependent variable as a predictor to partition the data into increasingly smaller groups. Any predictor can be used and reused to split new groups into nodes. (Strang, 2009, p. 5)

A variable may only have an influence on variable(s) on its right-hand side in the model. Ben (2010) offers further pertinent advice on how to read the diagram:

“In a path diagram, a direct effect is represented by a single-headed arrow between the variables concerned. A direct effect between two variables means that if one variable is changed, a change in the other variable is expected” (2010, p. 348). “A two-way arrow between two

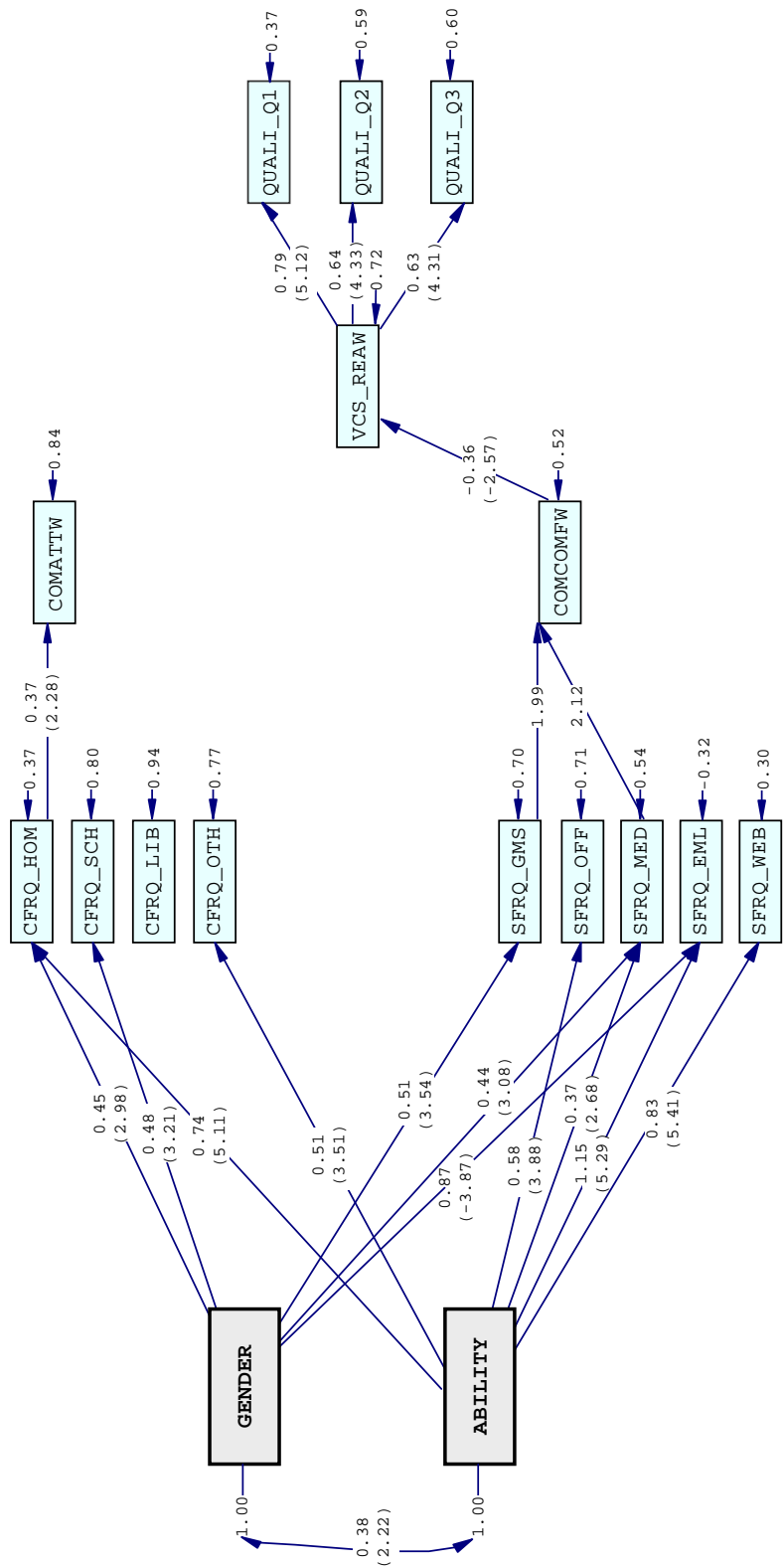
variables indicates that these variables may be correlated without any assumed direct relationship” (2010, p. 347).

The numbers included on the paths in the diagram represent the standardised estimates and t -values (enclosed in parentheses). As pointed out by Jöreskog and Sörbom (2001, p. 103), a t -value between -1.96 and 1.96 is not significant. Paths that fell inside this range were therefore removed from the model. The p -value for this model is $p < .001$, which means that there is less than 1% probability that a path is occurring by chance. Suhr (2000, p. 5) provides the following guide for estimating the direct effect of the path coefficient (standardised estimates):

- < 0.10 , ‘small’ effect
- ~ 0.30 , ‘medium’ effect
- $> .50$, ‘large’ effect

Thus, the path diagram serves two purposes; 1) it illustrates the paths where there exist a relationship between a predictor and indicator(s) and provides the significance (t -values, at $p < .001$) of the effect, and 2) the amount of influence the predictor has on the indicator(s) as determined by the standardised estimates. A tabular presentation of the path diagram is presented in Table 10.2. In the table (and in Figure 10.3) two path coefficient values can be observed with values greater than or less than one. This does not, however, mean that something is wrong. As Jöreskog (1999) points out, it is a common misunderstanding that standardised estimate coefficients must be less than one. This is also true for values less than one (Gartside and Heales, 2006).

The regression coefficients (which are included in the regression equations further down) indicate the amount of influence a change of one unit in a variables value would have on the next. Gorst et al. (1999, p. 5) provides an example where the value of a coefficient is 0.5; in this case, a ‘1’-point rise in the influencing variable (predictor) would lead to a rise of ‘0.5’ in the variable being influenced (indicator). The significant ($p < 0.001$) PA regression equations for the VCS (generated by LISREL) will now be discussed. Table 10.3 provides a reminder of the variables used in the model.



Standardised Estimates and (*t*-values), $p < .001$

Figure 10.3.: Confirmed Theoretical Model (N=165)

Table 10.2.: Standardised Estimates and *t*-Values (*t*-values in parenthesis)

	CFRQ_HOM	CFRQ_SCH	CFRQ_OTH		
GENDER	0.45(2.98)*	0.48(3.21)*	-		
ABILITY	0,74(5.11)*	-	0.51(3.51)*		
	SFRQ_GMS	SFRQ_OFF	SFRQ_MED	SFRQ_EML	SFRQ_WEB
GENDER	0.51(3.54)*	-	0.44(3.08)*	0.87(-3.87)*	-
ABILITY	-	0.58(3.88)*	0.37(2.68)*	1.15(5.29)*	0.83(5.41)*
	CFRQ_HOM				
COMATTW	0.37(2.28)*				
	QUALI_Q1	QUALI_Q2	QUALI_Q3		
VCS_REAW	0.79(5.12)*	0.64(4.33)*	0.63(4.31)*		
	SFRQ_GMS	SFRQ_MED			
COMCOMFW	0.41(1.99)*	0.49(2.12)*			
	COMCOMFW				
VCS_REAW	-0.36(-2.57)*				

N=165**p*<0.001

$$\text{CFRQ_HOM} = \beta_0 + 1.54(\text{GENDER}) + 3.04(\text{ABILITY}) + \text{error}$$

The above equation indicates that students' use of computers at home is predicted by GENDER as well as their perceived ability to use computers (ABILITY). GENDER was coded as '1' for female and '2' for male in the dataset. As pointed out by Ben (2010, p. 348), the coding standards are important indicators for interpreting the results. In the equation above, GENDER is signified to have a positive effect on CFRQ_HOM. Since male is coded with a higher value, the equation suggests that males use computers at home more than females (if a negative coefficient was observed, the opposite would apply). The equation also suggests that students who perceive themselves as having good abilities with computers are more likely to use a computer at home. The *t*-test values (2.98 and 5.11) reflect these findings.

Table 10.3.: Variables Used in the Path Model

Variable	Description	Comments
GENDER	<i>Student's Gender</i>	<i>F=1, M=2</i>
ABILITY	<i>Student's perceived ability to use computers</i>	
CFRQ_HOM	<i>Student's frequency of using computer at home</i>	
CFRQ_SCH	<i>Student's frequency of using computer at school</i>	
CFRQ_LIB	<i>Student's frequency of using computer at the library</i>	
CFRQ_OTH	<i>Student's frequency of using computer other places</i>	
SFRQ_GMS	<i>Student's frequency of using games</i>	
SFRQ_OFF	<i>Student's frequency of using office software</i>	
SFRQ_MED	<i>Student's frequency of using multimedia software</i>	
SFRQ_EML	<i>Student's frequency of using an email client</i>	
SFRQ_WEB	<i>Student's frequency of using an Internet web browser</i>	
COMATTW	<i>Student's attitudes towards computers</i>	<i>Latent</i>
COMCOMFW	<i>Student's comfort with computers</i>	<i>Latent</i>
VCS_REAW	<i>Overall reactions to the VCS</i>	<i>Latent</i>
QUALI_Q1	<i>"How do you view the educational value of the VCS?"</i>	
QUALI_Q2	<i>"Do you believe the VCS is an effective means of training?"</i>	
QUALI_Q3	<i>"Do you believe the VCS generates a valuable learning experience?"</i>	

$$\text{CFRQ_SCH} = \beta_0 + 6.51(\text{GENDER}) + \text{error}$$

The equation above strongly indicates that males are more likely to use computers at school than females. This is also supported by the t -value (3.21).

$$\text{CFRQ_OTH} = \beta_0 + 0.86(\text{ABILITY}) + \text{error}$$

GENDER was not found to have an effect on computer usage at other places (e.g. laptop, Internet cafe, etc). ABILITY, however, is shown to have some influence. This is also supported by a significant t -value (3.51, at $p < 0.001$).

$$\text{SFRQ_GMS} = \beta_0 + 0.41(\text{GENDER}) + \text{error}$$

Frequency of playing games is also shown to be influenced by GENDER, specifically males. The coefficient is quite low, while the t -value (3.54), on the other hand, reports a significant relationship (at $p < 0.001$).

$$\text{SFRQ_OFF} = \beta_0 + 1.50(\text{ABILITY}) + \text{error}$$

ABILITY is the only predictor for frequency of using office software (e.g. Word Processing, Spreadsheets). A relatively large influence is observed by the coefficient as well as a significant t -value (3.88, at $p < 0.001$).

$$\text{SFRQ_MED} = \beta_0 + 0.35(\text{GENDER}) + 0.36(\text{ABILITY}) + \text{error}$$

Frequency of using multimedia applications (e.g. drawing, painting, video or graphics) is predicted by GENDER (males, t -value: 3.08) and ABILITY (t -value: 2.68), though to a moderate degree.

$$\text{SFRQ_EML} = \beta_0 - 3.21(\text{GENDER}) + 5.16(\text{ABILITY}) + \text{error}$$

A negative coefficient is observed for GENDER, which suggests that girls use email clients more often than boys. A negative t -value (-3.87) is also observed. Frequency of use is also strongly influenced by ABILITY.

$$\text{SFRQ_WEB} = \beta_0 + 0.99(\text{ABILITY}) + \text{error}$$

Use of an Internet web browser is not influenced by GENDER. ABILITY is a predictor, but not as strong as was the case for use of email clients. The t -value (5.41) suggests a significant relationship between the variables (at $p < 0.001$).

$$\text{COMATTW} = \beta_0 + 1.20(\text{CFRQ_HOM}) + \text{error}$$

Student's attitudes towards computers are suggested to be influenced by their frequency of using a computer at home (t -value: 2.28).

$$\text{COMCOMFW} = \beta_0 + 7.95(\text{SFRQ_GMS}) + 9.61(\text{SFRQ_MED}) + \text{error}$$

Comfort with computers is indicated to be very positively influenced by student's frequency of playing games (t -value: 1.99) and using multimedia software (t -value: 2.12). The positive coefficients suggest an indirect influence by males.

$$\text{VCS_REAW} = \beta_0 - 0.20(\text{COMCOMFW}) + \text{error}$$

A negative coefficient (and t -value: -2.57) is observed in the equation. This may suggest that students who are not so comfortable with computers (COMCOMFW) have an overall reaction to the VCS (VCS_REAW) that is more positive than those who are. The previous equation demonstrated that COMCOMFW is highly influenced by students who frequently use computer games (SFRQ_GMS) and

multimedia software (SFRQ_MED). It is reasonable to infer that the benchmarks these students have set for what constitutes ‘wonderful’, ‘satisfying’, ‘stimulating’, ‘powerful’ and ‘flexible’ software are high. These two variables could therefore have an indirect influence on VCS_REAW.

$$\text{QUALI_Q1} = \beta_0 + 0.090(\text{VCS_REAW}) + \text{error}$$

Students’ perceived educational value of the VCS (QUALI_Q1, which 81% rated as high or moderate in the survey) does not seem to be influenced much by their overall reactions to the VCS (VCS_REAW). The standardised estimate of 0.79, however, seems to indicate a ‘large’ effect if the guide advanced by Suhr (2000, p. 5) is followed. A large t -value (5.12) also suggests that there is a strong relationship between the two variables. This suggests that VCS_REAW has a significant influence on QUALI_Q1 (at at $p < 0.001$), although an increase in the coefficient will not make much impact in the influence.

$$\text{QUALI_Q2} = \beta_0 + 0.24(\text{VCS_REAW}) + \text{error}$$

A similar case is apparent between students’ perception of the VCS as an effective means of training (QUALI_Q2, which 78% rated as true or potentially in the survey) and their overall reactions to the VCS. Using the standard estimate of 0.64, however, the extent to which QUALI_Q2 is predicted by VCS_REAW can be described as ‘large’. Again, the t -value (4.33, at $p < 0.001$) indicates a significant relationship.

$$\text{QUALI_Q3} = \beta_0 + 0.065(\text{VCS_REAW}) + \text{error}$$

The students’ perception of the VCS being a valuable learning experience (QUALI_Q3 which ~80% rated as true or potentially in the survey) demonstrates the lowest influence of VCS_REAW (t -value: 4.31). Standard estimate is 0.63, thus high enough to be said as having a ‘large’ effect (Suhr, 2000, p. 5).

10.2.1. Limitations

With regard to the CFA of constructs in the VCS HCI Survey, Section 8.4.1 in Chapter 8 discussed limitations imposed by a small sample size. Since the theoretical model confirmed in this chapter assumes variables from this survey, these potential limitations should be taken into consideration.

Missing values were replaced by multiple imputation. While this technique is commonly used, Schafer, 1999 alerts that undesired results may occur (this has not, however, been observed in this study).

10.3. VCS Assignment: Summative Assessment in the STIC Course

Section 6.3.6 in Chapter 6 described the VCS Assignment, which was an assessed component in the STIC course that sought to extract students' understanding of terms from traditional STIC theory and how they might relate to specific VCS output (results). The assignment can be observed in Appendix Q.

In his discussion on mixed-methods designs, Creswell (2008) describes the explanatory method ('the two-phase model') and, indeed, the others, as consisting of two stages at most. On a timeline, this study has consisted of three distinct phases of data collection, e.g. the VCS STIC Questionnaire (QUALI), the VCS Education (QUALI) —and HCI Surveys (QUANTI) and the VCS Assignment. Some question responses to the VCS Education Survey, as has been highlighted in this chapter, were also quantified. Following the explanatory method, quantitative data is first collected, then qualitative data some time later. The qualitative data is used to help explain or elaborate on the results from the quantitative data (Creswell, 2008). The exploratory method is essentially the 'reverse' of the explanatory. The triangulation (or concurrent) method collects both types of data at the same time and merges the data to understand a problem. The embedded design is almost the same, except one form of data is used to support the other. This research has adopted and extended on concepts from all of these methods.

A response to Research Problem 3 involves data from the qualitative STIC Questionnaire, the qualitative (and quantitative) VCS Education Survey, the

10.4. Observations From the Assignments

quantitative VCS HCI Survey and the qualitative VCS Assignment. These data are useful in determining to what extent the VCS is useful in reshaping preservice teachers' mental model about teaching and learning, as well as advancing their pedagogical content knowledge:

1. The STIC Questionnaire provided an insight into education students' understanding about important factors in the classroom (preconceptions of the mental model).
2. The path analysis established what influences education students' perception of the usefulness, effectiveness and value of the VCS in their education programme. It was important to investigate these causal relationships as they may impose positively or negatively on the aims of the VCS and thus affect the usefulness of the VCS.
3. The qualitative analysis of the VCS Education provided an insight into the education students' 'updated' mental models of the interaction processes in the classroom. The survey afforded a wealth of information that demonstrated the students' abilities to connect theoretical and practical concepts from the VCS to the real world.
4. The VCS Assignment, which will be examined below, made extensive use of examples from the VCS which allowed education students to demonstrate their understanding of relevant theory, pedagogy and practice. The assignment was submitted for assessment less than three weeks after they were involved in the VCS trials.

Thus, the research design has provided a series of 'snapshots' in time that may be used to assess the VCS' influence on the education students' mental models, and the assignment is the last of these snapshots.

10.4. Observations From the Assignments

More information was output from the assignment than what could feasibly be analysed in this study. Six assignments were therefore selected completely at

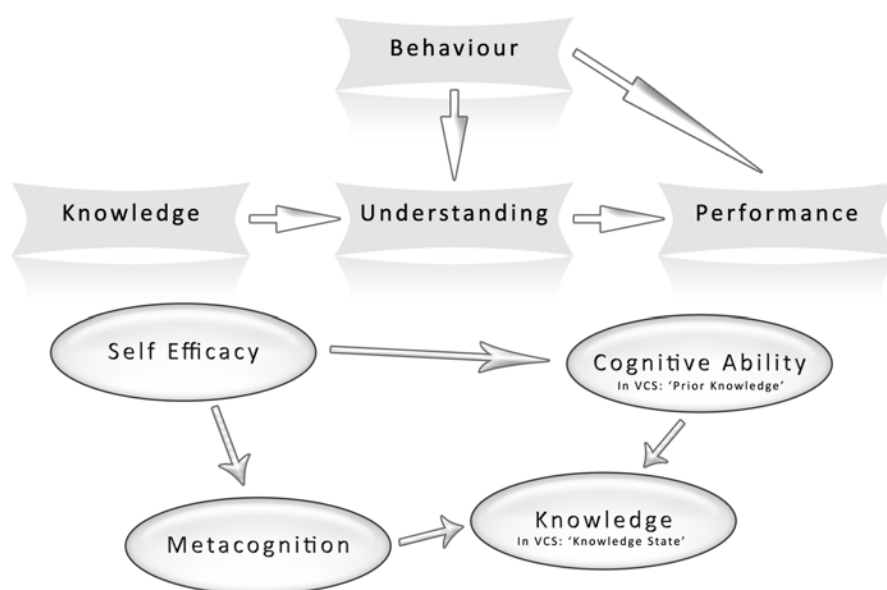


Figure 10.4.: STIC Assignment Diagrams

random, three for each gender, to examine the education students' responses. The assignment, which can be found in Appendix Q, asked of students to discuss terms related to STIC (Figure 10.4), how they are defined through literature and to add their own understanding of the interaction between all of them. It was further asked that students utilise two VCS result charts to make these connections and to justify their interpretation. A maximum word limit of 600 was imposed for the assignment.

10.4.1. ST0058 (Male)

ST0058 provides elaborate connections between the diagrams, charts, theory, the VCS and the individual virtual students. Apposite references to areas such as socially constructivist skills, scaffolding, behaviour management and Piaget's theory of schemes are all used as precursors to his detailed discussion about the VCS.

The Virtual Classroom Simulation (VCS) builds on the nexus between self-efficacy, cognitive ability, metacognition and knowledge.

10.4. Observations From the Assignments

The virtual student Kennet is used as a case scenario where ST0058 provides an understanding of his behavioural and cognitive attributes by linking these to the result charts and literature about social-emotional learning. Virtual student Annelie is discussed in a similar manner, and ST0058 makes pertinent comparisons with her classmate Kennet to demonstrate how their differences affect the learning outcome. One critique that can be made is that the student did not sufficiently demonstrate his understanding of metacognition. ST0058 provides the following concluding remarks:

The findings of the VCS have important implications for teaching theory and my own personal application during practicum. While still in a rudimentary form, the VCS is particularly effective for conceptualising performance among different types of students, and demonstrates the linkages between the facets of extrinsic and intrinsic behaviour that affect learning.

In summary, the student demonstrated a very good insight into the theory and managed to connect a number of these to the VCS students and the associated results charts. It is evident that the VCS provided an opportunity to relate theory with what was observed in the simulation and vice versa. The student's assignment can be found in Appendix W, Section W.1.

10.4.2. ST0159 (Female)

ST0159 begins by explaining (supported by research) how the two diagrams ('models') that pertain to traditional STIC theory (included in the course assignment) define knowledge differently. She then illustrates how one of the models is better connected to the VCS scenario:

Model 2 can be seen in the VCS graphs as teacher input (shown by events) promotes different self-efficacy responses in Annelie and Kennet.

An account of how the intrinsic variables of VCS students (e.g. self-efficacy, distractibility, cognitive abilities, etc.) interact in the context of learning is provided.

In Model 2 these interactive processes are the key to understanding. This is very different to Model 1 - where the emphasis of understanding is on the ability to perform. [...] The Models place different interpretations and emphasis on [knowledge, behaviour and understanding] as shown in the VCS graphs examples.

She also produces insight on metacognition, classroom management and contemplates on what the VCS charts may have looked like had the duration of the lesson been longer than five minutes.

In summary, ST0159 exhibited a number of good observations and substantiated these with references to literature as well as the VCS charts. All the main variables of the VCS student were taken into consideration in her discussion. The student's assignment can be found in Appendix W, Section W.2.

10.4.3. ST0087 (Male)

ST0087 opens the discussion by making the distinction between knowledge and understanding (with reference to literature). He then discusses behaviour, how it is related to the environment and ultimately affects performance, before an investigation into the learning process. A step-wise scenario for solving a problem is put forward that first involves self-evaluation in establishing a link between prior knowledge and the problem. The next step involves the immediate impact this initial evaluation of self and the problem may have on self-efficacy and the subsequent strategy, which ST0087 links to metacognition:

Furthermore, metacognition has been linked to cognitive ability (intelligence) as well, Sternberg refers to these executive processes as "metacomponents".

Kennet and his performance chart is brought into the discussion, and an applicable description is provided with regards to his self-efficacy and knowledge level:

Therefore we can see the evidence which suggest that kennet's knowledge and metacogniton are all below average. Overall, the VCS graph strongly support that there is a link between self efficacy, metacognition, cognitive ability and knowledge state.

In summary, ST0087 provided a different, yet interesting approach to the problem by telling a ‘story’ that connected an important argument from beginning to end. In the process he managed to address a number of factors pertinent to the VCS. The student’s assignment can be found in Appendix W, Section W.3.

10.4.4. ST0148 (Female)

In her introduction, ST0148 connects good behaviour and knowledge with understanding, how understanding gives meaning to knowledge. Good and bad behaviour, she says, impact performance (“showing what the student has learnt and is the effect of instruction”) — indirectly (through understanding) so does knowledge. This discussion sets the stage for a discussion about VCS students Kennet and Annelie. ST0148 provides some insight into their ‘personalities’ as well as observations from the charts, but she does not convincingly share her understanding of these observations. The final section does, however, address all the major variables pertinent to the VCS students.

If a student’s cognitive ability is high or the student wants to improve their ability, the amount of knowledge acquired increases due to activation in cognition. Similarly if a student has a high metacognition and has the skills on how to learn knowledge they can self regulate and change their learning style so that knowledge acquisition is increased.

In summary, ST0148 used a large number of references to literature to support in her discussion, which would typically be a positive. However, the assignment falls a bit short of demonstrating understanding through discussion and connecting with the VCS. This assignment appears to be written by a student whose first language is not English, which, if this is the case, may explain the lack of ‘flow’ in the discussion. The student’s assignment can be found in Appendix W, Section W.4.

10.4.5. ST0059 (Male)

ST0059 takes the approach of segmenting the assignment response into sections in which each term is addressed. With support from literature, he provides a good account on self-efficacy and then connects the established description with the VCS:

The graphs show little change in 'knowledge state' and 'metacognition' over the time period, but self-efficacy changes quite dramatically, which suggests that this factor is largely influential for creating positive student outcomes.

The student gives similar descriptions and references to VCS for metacognition, cognitive ability, knowledge, behaviour, understanding and performance and makes good use of literature. On the subject of knowledge, ST0059 demonstrates a good insight into interacting variables:

As the diagram suggests, the 'knowledge state' is the coming together of metacognition and cognitive ability. It is the final stage in which content is actually taken in and understood by the student.

Further insight was afforded with regard to behaviour:

Behaviour is related to the factors 'self-efficacy' and 'distractibility' which appear on the VCS graphs. The graphs show that there is a direct correlation between how engaged a student is with the task at hand, and their feelings of self-confidence at their ability to complete said task. This suggests that behaviour management in the classroom (which includes keeping students engaged and on task) has a large impact on student outcomes ('understanding' and 'performance').

In summary, the student demonstrated that he understood the theory well and he was able to connect interacting variables and infer on likely outcomes. He also did well in connecting a number of observations with the VCS. The student's assignment can be found in Appendix W, Section W.5.

10.4.6. ST0139 (Female)

ST0139 introduces a number of variables that she substantiates in more detail later in her assignment:

Educational psychology theory allows us to understand the determinants affecting student learning and behaviour in the classroom.

10.4. Observations From the Assignments

The self-efficacy, prior knowledge and metacognitive levels of individual students affect the ways that, and ability of, students to gain new knowledge and understanding to then apply this new knowledge through certain behaviours (performance).

With appropriate use of literature, she gives a good overview of self-efficacy. She also provides the best account for metacognition in this sample of six assignments. ST0138 continues to cover cognitive ability, stages of learning, understanding, scaffolding and challenges to build knowledge. The VCS is then addressed, and she points out the exact levels of knowledge, metacognition and distractibility for Annelie and Kennet, and highlights the differences between the two.

It is interesting to note that Annelie's knowledge state does not change throughout the lesson. Kennet's rises toward the end. We might conclude that Annelie has not been challenged. Kennet has. He has been challenged and learnt some new information (knowledge).

The quote above shows a section of ST0139's apposite discussion around the students' traits and events from the charts. She does a good job in providing theoretically and practically sound reasons for these throughout. She concludes the discussion with an insightful comment (and recommendation) about the VCS:

There are myriad ways in which one could draw links between educational psychology literature and the VCS. Although there is no way to replace practical experience within the classroom [...] it does offer us a wonderful insight into 'real time' relations between the teacher and their students. Limitations are that questions are non-specific, but it does take into account the inner workings of individuals within a certain environment. Just like metacognition, we need to look behind what is immediately perceivable and find ways in which to enrich experiences for both students and teachers; it is a reciprocal relationship.

It was encouraging to note what the VCS had done to shift mental models/schema about classroom interaction. The student's assignment can be found in Appendix W, Section W.6.

10.4.7. Discussion

It should be acknowledged that the students did not accumulate all the knowledge demonstrated in the assignments from the VCS only. In preparing for this assignment it is more than likely (and often evident from the use of citations) that they acquired further understanding from relevant texts. However, as has been stressed in many segments of this thesis, the VCS does not constitute of a piece of software only; it makes part of a course integration that includes activities, discussion, reflection and, if used as intended, assignments like this. Its intention is to provide a context into which students may learn and demonstrate learning.

10.5. Summary

This chapter advanced to the final stage of the statistical procedures in the research design (Chapter 6); path analysis. The triangulation of quantitative and qualitative data was demonstrated and reported, and a number of causal relationships were evident in the path analysis. One of the more interesting findings, which will be discussed in the next chapter, was the relationship between students' overall reactions to the VCS and how they perceived the usefulness, effectiveness and value of the simulation.

Six submissions to the VCS Assignment were chosen by random and studied to learn more about the education students' knowledge and understanding of STIC theory, particularly with regard to the VCS.

What remains for this thesis are the inferences of findings, a discussion of the implications of study to a number of areas and a final conclusion.

Chapter 11.

Conclusion

Although simulation models need to imitate situations in such a manner that a learner can gain insight into the interaction of variables within that system, these situations do not need to be exact replicates. In fact, simplistic simulations can actually assist novice managers by focusing their attention on important variables.

(FEINSTEIN AND CANNON, 2002, P. 437)

The trans-disciplinary nature of this research has been emphasised and demonstrated in this thesis. Expert technology transfer from education, psychology, social sciences, conceptual modelling, computer science and underlying mixed methods research design have been instrumental in underpinning the research and development of the VCS. This study has designed and operationalised a computer simulation for its learning outcomes and this thesis has encapsulated the motivation, conceptualisation, theory, development, trials and evaluation involved in these activities.

The VCS was conceived to address a number of challenges in education and a research design was carefully developed to address three main problems of the study. Four interacting strands were explicitly involved in addressing the problems:

1. Education

- a) Practicum during pre-service teacher education is an important facet for gaining insights into lesson preparation, design, delivery and the interactions that happen in a classroom. Difficulties of locating

placements (schools) coupled with education students not gaining the full knowledge and skills in working effectively in the classroom directed this study. Hence, these background concerns and challenges provided insights into the objectives that motivated the problems and the expert knowledge of how they relate to teaching and learning in the classroom context.

2. Conceptual and Simulation Modelling (CSM)

- a) Instrumental in the deconstruct-reconstruct operations of the educational input and to enable the translation of a real system into identifiable components for advancing the simulation framework. Both the underlying theories in teacher-student-task interactions and the fundamentals of simulation and modelling facilitated the construction of the VCS.

3. Computer Science

- a) Design and development of a computer simulation software that integrated the abstracted system.

4. Measurement and Evaluation (M&E)

- a) Constructed an apposite research design to evaluate the outcomes from the disciplines above. It took into cognisance the richness in qualitative and quantitative data, and coupled it with actual students' work produced at the end of an education course, to present and highlight the use of simulation for learning and education.

The fourth discipline, which embodies Chapters 6 to 10 of this thesis, afforded procedures to operationalise the VCS and accumulate rich and complemented data from a population of education students. Detailed analyses of the data were instrumental in ascertaining tangible information to enable a grounded response to the research problems. Moreover, the discipline provided methods to demonstrate the reliability and validity of its own procedures, which authenticates the application of the simulation into preservice teacher education.

11.1. Research Problem 1 (RP1)

Problem 1. What are the most relevant entities in a classroom environment and what are their most significant attributes?

- How can these entities be reconstructed and combined into an efficient interaction model?

Chapters 2 and 3 involved the disciplines of Education and CSM. The disciplines provided expert input to address RP1 and to establish a foundation for the other research areas. This foundation was expounded and operationalised by the discipline of computer science (see Chapters 4 and 5). The output from the discipline, the simulation software, was exploited in user trials to validate the response to RP1.

11.1.1. Outcomes

A comprehensive database of constituents (entities and attributes) relevant to the classroom environment had been established in prior VCS-related research. All of these were carefully reassessed against the objectives of the current study. Three entities were characterised as essential to the objectives of the VCS; STUDENT, TASK and TEACHER (the latter being the user of the simulation). They were implemented in a classroom environment/setting also essential to the objectives.

The core attributes for the STUDENT and TASK entities are summarised in Table 11.1. A number of additional attributes were included in the entities' conceptual models to facilitate behaviour and interaction (see Section 3.4.2.2 in Chapter 3).

The selected attributes were determined as most fundamental to the classroom interaction processes that the simulation aims to demonstrate. Moreover, the constructs were key in substantiating important concepts taught in the relevant education course. The system abstraction process was mindful of the human entity (TEACHER), as it carried implications for the interaction model as well as design of the simulation software with regards to Human-Computer Interaction (HCI) and the Graphical User Interface (GUI).

Table 11.1.: Core Entity Attributes of the Conceptual Model

Entity	Attributes
STUDENT	Distractibility Metacognition Mood Prior Knowledge Self Efficacy
TASK	Difficulty Novelty

11.1.2. Research Findings

The research design did not put in place specific measures to directly validate or otherwise inform on the response to this problem. The informal VCS STIC Questionnaire (Section 7.3, Chapter 7) sought to elicit what the education students regarded as significant attributes in the classroom environment. The questionnaire served three main purposes:

1. Get students thinking about Student Teacher Interaction in the Classroom (STIC) before the introduction to the VCS.
2. Compile the responses, present an overview of findings back to the population, and, for reflection purposes, draw parallels between the education students' chosen attributes and those included in the VCS.
3. Use the data as a first snapshot of the population's understanding (mental models) of classroom interaction and pedagogies for teaching and learning and compare these with findings from data collected later in the research.

The questionnaire did not make reference to the VCS, nor did the questions accurately reflect that of RP1. It can be observed in the responses, however, that the education students did not demonstrate a deep understanding of concepts in the theory around classroom interactions. For example, the questionnaire was comprised of three questions, each question afforded five factors/attributes (related to teacher, student and task) to be entered and eighty-four students responded to

11.2. Research Problem 2 (RP2)

the questionnaire. With a potential of $3 \times 5 \times 84 = 1260$ factors/attributes, ‘self-efficacy’ was received in one response only. Metacognition was not mentioned at all. In contrast, students’ responses to the VCS Assignment (which was administered after the VCS trials) were able to highlight and discuss the impact of metacognition on interactions within the class and the impact on student learning. It was evident in the dataset that a large portion of the responses took features observable from the student’s perspective into consideration, not the teacher.

11.1.3. Comments

For the first time in VCS-related research, high-level CSM was utilised to reduce Task and Student attributes to a set of essential few for inclusion in a simulation model. The abstraction was informed — and validated — by established theories in education and advice from expert educators. The response to this problem carries multiple implications for the research and the remaining two problems. RP1 did not stipulate *how many* significant attributes to identify, nor the exact nature of the classroom interaction context. This will, however, be reflected in findings from RP2 and RP3.

11.2. Research Problem 2 (RP2)

Problem 2. Identify appropriate technologies (hardware and software) that combined allow for efficient and dynamic implementation, management and compatibility of the VCS with regard to:

- input/output (I/O)
- graphical user interface (GUI)
- stability
- speed
- usability

- extensibility
- reliability
- functionality.

What is stipulated in RP2 imposed careful consideration of the outcomes of RP1 and the objectives of RP3. Chapters 4 and 5, which provided a detailed account of the challenges and solutions involved with the development of the VCS software, represent the researcher's response to the problem. It is evident that both the Simulation Framework (SF) and Graphical User Interface (GUI) met all criteria stipulated by the problem.

11.2.1. Outcomes

The points below address the outcomes of the SF against the criteria in RP2:

- Input and output
 - The SF demonstrates a sophisticated conceptual and technical solution to the input and output requirements that is both efficient and scalable. In particular, the design and implementation of the simulation's Database Management System (DBMS) was a milestone achievement in this research. It came as a result of extensive research and a number of iterations in the development cycle.
- Graphical User Interface (GUI)
 - While the SF does not implement the GUI, it provides every feature required by the GUI to interact with the system. This is not to say that the SF is married to this particular VCS GUI. On the contrary, the SF was built from ground up in a modular way that renders a system that is detached from any single GUI, yet can accommodate for many.
- Extensibility, Stability and Reliability

- The modular and standardised design of the SF accommodates extensibility for new functionalities and features with ease.
 - The SF is detached from the GUI, as well as the simulation model, and can thus be used as a framework for simulations with operational needs similar to those of the VCS.
 - The design allows for modules to be tested separately, which eases and improves quality and validation testing of the system’s operations. Error handlers are heavily implemented in the source code and serve two main purposes. First, they ensure that the simulation can resume operation, even in the event of non-critical errors that would otherwise cause a system to ‘crash’. Second, the detailed logs of errors that they provide significantly ease the detection, troubleshooting and addressing of issues. This design was useful in the development stages of the system and has ensured that the system is both reliable and stable.
 - The VCS implementation is Java-based, a technology that supports a large number of platforms (i.e. ‘cross-platform compatible’). This means that the VSC compatibility extends to all Java-enabled systems (Java Standard Edition 6 or later). Appendix A.1 lists the operating systems on which the VCS was developed and tested.
- Speed
 - Software performance is determined by its design as well as the capabilities of the host computer system. Memory allocation and management for the VCS is performed by the Java Virtual Machine. The VCS includes an inbuilt meter that indicates its Random Access Memory (RAM) consumption. No excessive memory consumption has been witnessed.
 - Performance profiling of the VCS (using profiling tools included in the NetBeans IDE software) suggests that the software is not demanding in terms of processing power. The VCS has not been witnessed to be impeded by performance issues or to have impeded the performance of the host computer.

Chapter 11. Conclusion

- Delay is experienced when loading the VCS. This is imposed by the start up tasks of the DBMS, which require writing a large number of session-related files and folders on the host computer’s hard drive. Read and write operations on a hard drive are typically slow, and this is not a limitation of the VCS. Measures are nonetheless taken in the framework, which implements logic for feeding the GUI splash screen information. This is a feature implemented specifically for the user to keep them informed about the start-up progress.

- Functionality
 - The SF has demonstrated that it meets all requirements of the simulation objectives. No desired features had to be discounted due to limitations in the SF. Moreover, the system is in its current state very capable of meeting the needs of features planned for future development (thus it is scalable).

The GUI is the communication layer between the user and the SF. A number of considerations had to be made to accommodate communication in the best possible way. The GUI provides an attractive, standardised, logically segmented and easy to use display that is user-centric. It provides a number of different modes for user input, as well as presentation output. It implements a number of measures to keep the user informed (in-built information, alerts, auto-open results, etc.) and restrict the user from doing the ‘wrong thing’ (e.g. error-checking of user input, logical blocking of input at certain times/events, confirmation for critical actions such as stop/exit the simulation, etc.). The GUI also takes into account the pedagogical objectives of the research through flexible display of information. Although the graphical representation of students superseded original expectations of what could be achieved, it is nonetheless an area that needs further work. Overall, however, the GUI has been very successful in addressing the criteria of the research problem. The next section will examine findings extrapolated from the research design, and how these validate the researcher’s response to RP2.

11.2.2. Research Findings

The above responses to RP2 were given from a software developer’s perspective, and they take a number of different aspects into consideration. However, the developer is not an education student. Moreover, the developer’s experience, insights and bias towards the system makes him the least suitable person to suggest how appropriate the solution is to the user. The User Interaction Satisfaction Scale of the VCS HCI Survey was therefore implemented, in conjunction with the VCS trials, for its suitability to extrapolate system-specific feedback to address this problem from the users’ point of view. Chapter 9 describes the User Interaction Satisfaction Scale and its question in detail.

Statistics from observed data will be presented to address the different aspects of the software. Every question in the scale utilised the Likert scale with 4 response categories (negative→positive) and an ‘NA’ option (not applicable). Table 11.2 provides an overview of the statistics, where the *Nr* column indicates number of students who responded to the question with a value (e.g. 1, 2, 3 or 4). ‘NA’ responses were not included when calculating percentages, which explain the fluctuating value in the *Nr* column. The value 1 is always the most negative and value 4 is always most positive. Thus, a higher value indicates a more positive response to the VCS.

Table 11.2.: User Interaction Satisfaction Scale— Summary of Responses

	1	2	3	4	<i>Nr</i>
Ease of Use (difficult → easy)	-	5.3%	38.2%	56.6%	76
Readability (hard to read→easy to read)	-	5.0%	13.8%	81.3%	80
Screen Layouts Helpful (not at all→very)	-	6.0%	49.4%	44.6%	83
Help/Info/Error (inconsistent→inconsistent)	-	2.6%	40.8%	56.6%	76
Help/Info/Error Clarity (confusing→clear)	-	5.2%	44.2%	50.6%	77
Kept Informed? (never→always)	-	6.0%	42.9%	51.2%	84
Error Messages Helpful? (unhelpful→helpful)	-	2.9%	70.6%	26.5%	34
How Difficult to Learn VCS? (difficult→easy)	1.2%	8.4%	14.5%	75.9%	83
Tasks Straight-forward? (never→always)		6.0	25.3	68.7	83
Speed of VCS (too slow→fast enough)	7.1	10.7	32.1	50.0	84
Reliable (very unreliable→very reliable)	2.5	6.2	25.9	65.4	81

Nr=Number who responded / N=86

Chapter 11. Conclusion

- *Overall Reactions to the Virtual Classroom Simulation (difficult / easy)*
 - ~**95%** responded in categories 3 or 4 (positive end of the scale)
- *In terms of readability, how did you find the text on the screen?*
 - ~**95%** responded in categories 3 or 4 (positive end of the scale)
- *Were the screen layouts helpful?*
 - **94%** responded in categories 3 or 4 (positive end of the scale)
- *How would you rate the help, information and error messages that appeared in the program?*
 - *Inconsistent / Consistent:* ~**98%** responded in categories 3 or 4 (positive end of the scale)
 - *Confusing Clear:* ~**95%** responded in categories 3 or 4 (positive end of the scale)
- *Did the program keep you informed about what it was doing?*
 - ~**94%** responded in categories 3 or 4 (positive end of the scale)
- *Did you get any error messages? If so, how helpful were they?*
 - ~**97%** responded in categories 3 or 4 (positive end of the scale)
- *How difficult was it to learn to use the VCS?*
 - ~**90%** responded in categories 3 or 4 (positive end of the scale)
- *Could the tasks in the program be performed in a straight-forward manner?*
 - **94%** responded in categories 3 or 4 (positive end of the scale)
- *How was the speed of the VCS?*
 - **82%** responded in categories 3 or 4 (positive end of the scale)
- *How reliable was the program?*
 - ~**91%** responded in categories 3 or 4 (positive end of the scale)

11.3. Research Problem 3 (RP3)

The output above clearly shows that the vast majority of the students were positive to all aspects of the software. The only item with a percentage score below ninety concerned the speed of the VCS, but even this feature was rated positively (at 82%). However, the speed-factor is an important consideration of the VCS or related simulations that are run online for learners undertaking a distance education course.

RP2 was concerned with the features that the system should accommodate for in terms of the user and modules in the software. To the end user, this presented itself as usability. The researcher's response to this problem was substantially validated by the population.

11.2.3. Comments

VCS development employed only software technologies that were obtained either from open source or community-based repositories. Moreover, design and programming of the VCS software was performed on a computer with modest, consumer level, specifications. The research highlights that sophisticated ICT solutions for education can be produced on a minimal budget, with standard equipment, without compromise to quality and pedagogical details.

11.3. Research Problem 3 (RP3)

The research design served two main purposes in regard to this problem. First, it facilitated the integration and operationalisation of the VCS in a teacher education course with the aim of advancing education students' mental models. Second, it implemented a number of measures to examine how successful the VCS was in achieving this aim. A final reminder of Research Problem 3:

Problem 3. To what extent is the VCS useful in reshaping preservice teachers' mental model about teaching and learning, as well as advancing their pedagogical content knowledge?

- How can this be accurately measured?

- How can the impact of the VCS be further improved?

The procedures for addressing the problem were outlined in the last chapter:

A response to Research Problem 3 involves data from the qualitative STIC Questionnaire, the qualitative (and quantitative) VCS Education Survey, the quantitative VCS HCI Survey and the qualitative VCS Assignment. These data are useful in determining to what extent the VCS is useful in reshaping preservice teachers' mental model about teaching and learning, as well as advancing their pedagogical content knowledge:

1. The STIC Questionnaire provided an insight into education students' understanding of important factors in the classroom (preconceptions of the mental model). This parallels the pre-test for this study, and sought to gauge student's initial mental model/schema and understanding of the factors/attributes interacting in a classroom.
2. The path analysis established what influences education students' perception of the usefulness, effectiveness and value of the VCS in their education programme. It was important to investigate these causal relationships as they may impose positively or negatively on the aims of the VCS and thus affect the usefulness of the VCS.
3. The qualitative analysis of the VCS Education provided an insight into the education students' 'updated' mental models of the interaction processes in the classroom. The survey afforded a wealth of information that demonstrated the students' abilities to connect theoretical and practical concepts from the VCS to the real world.
4. The VCS Assignment, which will be examined below, made extensive use of examples from the VCS which allowed education students to demonstrate their understanding of relevant theory, pedagogy and practice. The assignment was submitted for assessment less than three weeks after they were involved in the VCS trials. The final assignment provided insights to the learning that had

occurred through the use of the VCS. It parallels a post-test to gauge changes to mental model/schema.

11.3.1. Research Findings

A theoretical model was constructed and its constituents (instrument scales) were validated. The path analysis found evidence to suggest that students who are very comfortable with computers, particularly those who frequently play games and use multimedia technologies, are not as likely to have positive reactions to the VCS as those with less computer comfort. Probably, the VCS was not as interactive as some of the computer games these students were playing. This could imply that educational applications and simulations (VCS in this study) have to ‘catch-up’ with gaming software to engage and be attractive to learners. The analysis also showed that there was a strong *relationship* between the overall reactions to the VCS and the populations’ view of the Usefulness, Effectiveness and Value (UEV) of the simulation for education. The regression coefficients indicated that the overall reactions to the VCS had little *influence* on the perceived UEV. This may suggest that the group of students who showed less positive reactions to the VCS still found UEV in the simulation. It could be explained by the fact that users with high expectations to interactive digital content still recognised the utility and usefulness in the specific-purpose product. Further research would be needed to establish this with certainty. It is nonetheless evident that education students’ attitudes towards — and experience with — computers do not have a significant influence on their overall reaction and perceived UEV of the VCS. Identifying the presence (or non-presence) of relationships between the variables and factors provided valuable detail to this study’s results. Without the techniques employed in the current research, there would have been little information available to establish the predictors for VCS uptake.

It was apparent from responses to the STIC Questionnaire that education students had fixed views about constructs pertaining to teaching and learning. The VCS Education Survey, which was administered not long after the questionnaire, displayed a completely different set of thinking and engagement with the content (Section 10.4 in Chapter 10). The STIC Survey was not designed

Chapter 11. Conclusion

to elicit as much information as the VCS Education Survey. The STIC Survey was also administered under different circumstances (e.g. paper-based, no VCS stimulus). Nonetheless, it is evident in the VCS Education Survey that there was a shift in the students' focus. This was also observable by the administrator of the trials and surveys. A number of students learnt at least one new term and the meaning of that term (Section 7.4.3.9 in Chapter 7). Many more were able to for the first time apply, observe, challenge and reflect on their understanding about education theory and practice. It is not unlikely that respondents who rated the VCS as having little or no UEV received some value as well. Sometimes learning happens, or existing knowledge is reinforced, when existing knowledge is employed to challenge or otherwise try to disprove other knowledge. Cognitive dissonance played a crucial role in reshaping student's mental model/schema about classroom interactions.

The final snapshot of the education students' mental models was afforded by the VCS Assignment. The analysis of student responses provided encouraging observations. Again, it was apparent that a change of mental models had enhanced the level of critical insight and ability to connect the VCS to education theory and vice versa. Lectures, tutorials, readings, preparation and the fact that the assignment was assessed should, of course, not be disregarded. These factors more than likely influenced achievements in the assignments to a greater extent than the VCS. However, the VCS does provide an alternative perspective, one that requires the student to put content into context and to challenge their mental models about teaching and learning. This involves the second chapter's discussion with regard to theory, reflection, practice and praxis.

As has been more or less alluded to in the latter part of this thesis, the findings from this research strongly indicate that the VCS has the capacity reshape preservice teachers' mental models about teaching and learning, as well as advancing their pedagogical content knowledge to a great extent. This statement is made on the premise that the VCS is soundly integrated in a course, preferably to a greater extent than what was allowed for in this research study. This partly addresses the embedded problem of "*How can the impact of the VCS be further improved?*" The other part of the response involves further research and development of the software, as well as supporting mechanisms (e.g. materials and guidelines) that ease

the uptake and integration of the VCS in institutions for teacher education.

11.3.2. Concluding Remarks

It is evident that this study has, through its substantial progress, addressed all three research problems. Although there was an expectation that the VCS would have some merit, the reason for conducting the study was not to prove the expectation right. The research was undertaken to learn *whether or not* a unique simulation for education like the VCS would have a place in education. Thus, the successfulness of this research should not be measured by the positive results of the VCS, but rather on the study's successfulness in responding to a set of problems to discover new knowledge. Had the outcome for the VCS been less favourable, the research would nonetheless contribute the same amount of knowledge about something that was not previously known.

11.4. Implications of the study

The contribution of knowledge about the usefulness of the design and trial of a simulation software in education is not the only advancement of this research. A model for the use of technology and simulation to enhance learning has been offered. Simulations offer great opportunities to connect theories into praxis and the feedback through the various questionnaires and insights through students' work provide a basic model for learning (Figure 11.1) . Moreover, the research design of this study has demonstrated rigour by following and incorporating modern evaluation and measurement theories and techniques.

11.4.1. Theoretical Implications

The Virtual Classroom Simulation highlights a number of attributes that operate in the classroom. These numbers are likely to grow in size in future work. The study has demonstrated the conceptual modelling process, by which an abstracted representation of a real system (classroom interaction) can be specified and made operational. A system has been developed that allows the user to interact, observe,

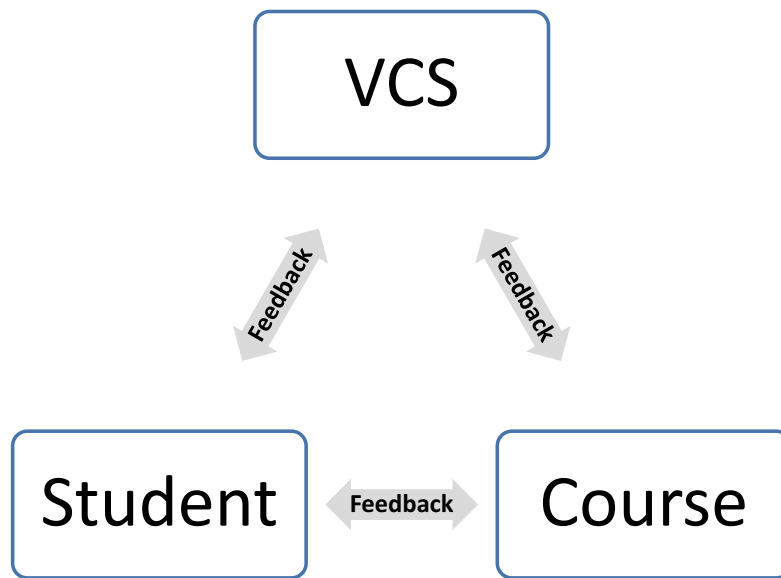


Figure 11.1.: VCS Feedback Cycle

reflect, discuss and connect theory with practice. The VCS is more than a software system; if implemented in a pedagogical framework, it is also a system for directing praxis. In real practice, more time should be invested in the system than what was afforded by this study.

11.4.2. Methodological Implications

The software design has rounded a number of milestones and the research has demonstrated the benefits of cross-fertilisation between disciplines. However, it is not implied that this study has reached a level of unsurpassed technical sophistication. Nonetheless, the act of bringing together various technologies to build a system like the VCS is not a simple activity involving the stitching together of some computer codes. The achievement lies in making the process explicit and documenting it so that others can learn from it. In this study extensive research provided no ready-made ‘recipe’ for developing a simulation like the VCS. Thus, a new solution was developed for this particular research, and several technologies were sampled to find the best combination to address the objectives. The methodology has contributed to the *processes* for simulation

design in education.

The research design provided the means for responding to the three research problems in this study, as well as measuring the features of the software and its ability to meet set objectives. Both qualitative and quantitative methods were implemented for collecting data. Data was obtained using traditional as well as digital modes (online survey). The research has demonstrated how modern evaluation and measurement techniques were used to establish the collected dataset's reliability and validity. It has also demonstrated the use of imputation methods for handling missing data.

11.4.3. Implications for Simulation Design

This research has demonstrated the feasibility of education-based solutions utilising Information and Communication Technologies (ICT). The VCS is the first of its kind and, as such, the first to be trialled and integrated in a real education context to assist preservice teachers prepare for their practicum. The VCS is a specific-purpose software for a specific user group. The Human-Computer Interaction (HCI) aspect was therefore given careful attention, particularly with regards to the GUI. The system was built from first-principle, on consumer-level equipment, utilising open source technologies. It is portable and can operate on computers across operating systems.

Many of the simulation's operations are implicitly captured in the results charts, as these are the result of the sophisticated Bayesian algorithms that are used to simulate the students' behaviour. While it has been established that human behaviour cannot be accurately modelled, this research has worked on the premise of likely output. The current algorithms provide a platform for future refinement and expansion.

An important output from the simulation design is the developed framework. The framework has properties that make it attractive for re-use in simulation projects with operational requirements similar to that of the VCS (e.g. real-time, interactive human-in-the-loop, discrete -and continuous). It would be particularly suitable in interactive simulations for education and training.

11.4.4. Implications for Practice

The VCS may be thought of as a director of praxis and a catalyst for reflective practice as it establishes the nexus of educational theory, hands-on practice and reflection. A case was argued for the need for the VCS, e.g. the number of challenges in locating placements for pre-service teachers and for exposing them to the totality of experience in classroom interactions.

The VCS Feedback Cycle introduced in Figure 11.1 on page 294 suggests the continuous bidirectional feedback that occurs between the student and the course, the student and the VCS and the course and the VCS. The model suggests how the VCS provides a new approach for the transfer of knowledge and praxis through the interrogation of content, practice of concepts and reflection. These aspects of teacher education are important precursors for practice.

11.4.5. Implications for Further Research

Chapter 1 established that this research was a pilot study. The VCS software went through one cycle of trials, and it highlighted a number of areas pertinent for future development. Further research is also guided by feedback from the population, which provided a substantial amount of data. The research design ensured to elicit information specifically pertaining to further research (e.g. “*What improvements need to be incorporated into the VCS as a teaching and/or training tool?*”). One readily made observation is that educational applications and simulations (VCS in this study) may need to ‘catch-up’ with gaming software to engage and be attractive to learners.

The VCS Feedback Cycle in Figure 11.1 on page 294 provides a model that opens up for a new area of research that was only partially explored in the current study. The research has also opened up for further research and development in the area of high-level simulations for education, with consideration for the added value of a trans-disciplinarity approach.

11.4.6. Limitations of the Study

The School of Education at the University of Adelaide should be acknowledged for allowing access to the population. The generous cooperation from the School of Education stretched the limits of how much time can be afforded to a research study. While the research worked well within these limitations, there was not enough time to satisfactorily conduct both the trials and surveys. The VCS trial was an essential part of the research, but the allocated time was strictly limited to accommodate for the surveys, an equally important activity. A number of students were therefore unable to complete one or both surveys. A number of survey responses also indicated that the students would have liked to spend more time with the VCS.

The study has also stressed the importance of VCS integration. While to some extent this was achieved in the study, it does not represent the extent of integration that is envisaged in real practice. Further integration and more time and access for evaluation would provide better measures for the impact of the VCS.

A limitation was apparent in the design of the User Interaction Satisfaction Scale, part of the VCS HCI Survey. The horizontal layout for the Likert-scale placed the 'NA' (not applicable) option too close to the other response categories. This may have caused some confusion between the right-most value of the scale and the 'NA' option and affected the accuracy of the data.

Rather than perceiving a limited budget as a limitation of the study, this research embraced the opportunity to investigate what can be achieved with open-source technologies. It is therefore a contribution, not a limitation.

11.5. Concluding Remarks

The design of the Virtual Classroom Simulation and the triangulation of data through a mixed methods research design highlight the complexities inherent in transdisciplinary research studies. The processes and directions adopted in this study provide crucial how causation in education could be explored *Morrison (2009)* and also how those keen in education need to work in collaboration. As highlighted in the demands of the OECD 2007 and OECD 2009 reports, innovators in ICT4E (Information and Communication Technologies for Education) have to

Chapter 11. Conclusion

work closely with colleagues in the field of education and educational research to address issues to enhance and optimise learning (be it for students in schools or for prospective teachers of these students).

Moreover, it is pertinent that educators play a key role in the design of applications and simulations for education and their evaluation. This critical evaluation need careful triangulation of data, and must include evidence of students' learning demonstrated through measurable outcomes.

This study recommends that the VCS is researched and developed further with the aim of making the system available to all providers of teacher education in Australia. This thesis answers the call asked for by *Morrison (2009)*.